Experiments in Multiple-Baseline Stereo

Tomoharu Nakahara and Takeo Kanade

August 1992

CMU-CS-93-102

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890

This research was sponsored by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, Ohio 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597.

keywords: stereo, 3-D vision, range sensing, multi-image fusion

Abstract

This paper presents the results of our experiments with a unique multiple-baseline stereo technique. This algorithm for producing precise, unambiguous depth maps from a set of multiple stereo pairs was developed by Okutomi and Kanade. Early versions of the algorithm were shown to perform well under controlled conditions in the Calibrated Imaging Laboratory (CIL).

In this paper, the algorithm is further applied to complex outdoor scenes with variable lighting conditions and large depth ranges. While Okutomi and Kanade used stereo pairs acquired by moving a camera horizontally, we also investigated the use of stereo pairs taken by moving a camera in both horizontal and vertical directions. The use of stereo images with two orthogonal baseline orientations removes ambiguity and increases precision without the problems associated with the orientation of the features in a scene. We also show that the shapes of the sum of squared-difference (SSD) curve indicates the reliability of the match, and suggest a method to classify matches into various types and to improve estimates when a false match occurs. Finally, results are presented to show the effectiveness of this algorithm and the classification method.

1. Introduction

In stereo matching, a longer baseline gives a precise depth estimate, because the depth is calculated by the triangulation method. A longer baseline, however, poses its own problem in matching. A longer disparity range must be searched, some parts in a scene may be occluded, and the appearance of some objects in the scene may change significantly between images. Matching becomes more difficult, and there is a greater possibility of false matches. Conversely, a shorter baseline makes matching easier, but reduces the precision of an estimate. There is a trade-off between precision and correctness.

Our multiple-baseline stereo technique was developed by Okutomi and Kanade to solve this problem [OK91]. This method uses multiple stereo pairs with different baselines generated by a lateral displacement of a camera. Matching is performed simply by computing the sum of squared-difference (SSD) values between multiple stereo pairs. The SSD functions for individual stereo pairs are represented with respect to the inverse distance, and they are simply added to produce the sum of the SSDs. This resulting function is called the SSSD-in-inverse-distance. The range estimate is calculated by finding the minimum of the SSSD-in-inverse-distance curve. This curve shows a unique and clear minimum at the correct matching position even when the underlying intensity patterns of the scene includes ambiguities or repetitive patterns.

This paper presents results obtained by this algorithm when applied to outdoor scenes with complex lighting conditions and large depth ranges. While Okutomi and Kanade used stereo pairs acquired by moving a camera only horizontally, we use stereo pairs taken by moving a camera in both horizontal and vertical directions. Taking stereo images with two orthogonal baseline orientations removes ambiguity and increases precision without the problems associated with the orientation of the features in a scene. We also show that the shapes of the SSD values near the estimate indicate the reliability of the match, and suggest a method to classify matches into four types: a good match and three kinds of false matches. For some of the detected false matches, we show a method to improve the estimates using only reliable SSDs.

In the next section we briefly describe the multiple-baseline stereo algorithm, the method to detect and to classify false matches, and a method to improve the estimates for some of the detected false matches. Section 3 describes the implementation and usage of the several programs which we have developed. Section 4 provides many experimental results with horizontal baselines and with combined horizontal and vertical baselines. The results of detected false matches and corrected estimates are also shown. Section 5 presents our data acquisition and calibration procedures.

2. The theory of multiple-baseline stereo

2.1. The basic algorithm

The multiple-baseline stereo method uses multiple stereo pairs with different baselines. These stereo pairs are taken by cameras at positions $P_0, P_1, \ldots, P_n$ along a line with their optical axes perpendicular to the line. The resulting set of stereo pairs have different baselines $B_1, B_2, \ldots, B_n$ as shown in fig. 1. In stereo matching, the SSD function is a useful and popular method for finding correspondences. Let $f_0(x, y)$ and $f_i(x, y)$ be the image pair at the camera positions P_0 and P_i. The SSD function over a window W at a pixel position (x, y) of image $f_0(x, y)$ for the candidate disparity $d_{(i)}$ is defined as

$$SSD(x, y, d_{(i)}) \equiv \sum_{k \in W} \sum_{l \in W} (f_0(x+k, y+l) - f_i(x+d_{(i)}+k, y+l))^2, \quad (1)$$

where the $\Sigma_{k \in W}$ and the $\Sigma_{l \in W}$ indicate summation over the window. Our multiple-baseline stereo technique uses inverse distance instead of the conventional disparity. The inverse distance ζ is

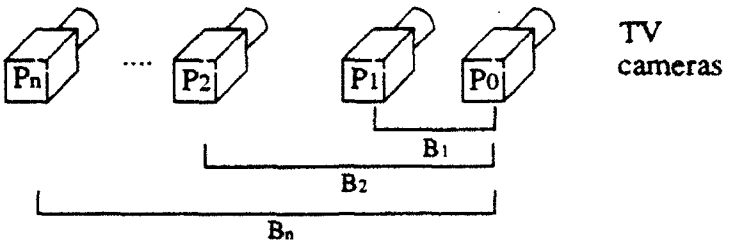


Fig. 1: Camera positions for stereo

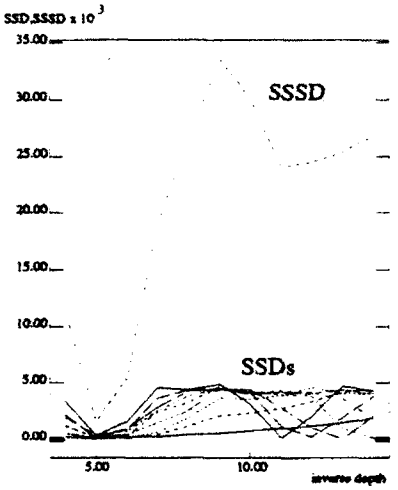


Fig. 2: SSD and SSSD values vs. inverse depth

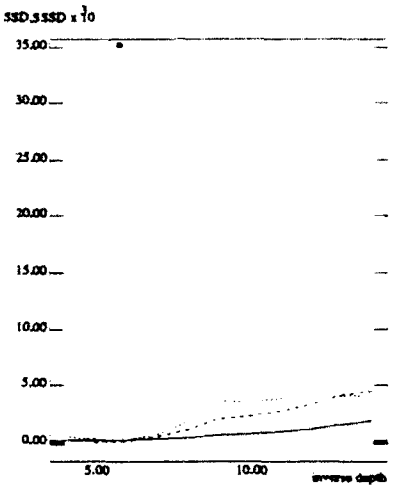


Fig. 3: SSDs of shorter baselines

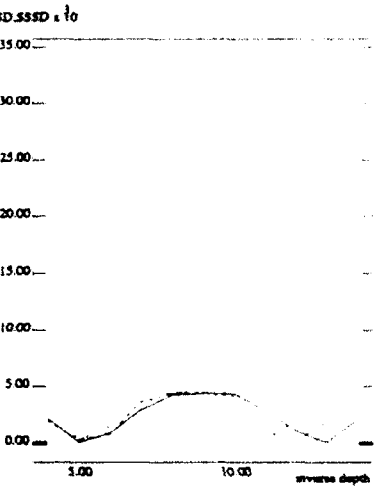


Fig. 4: SSDs of longer baselines

$$\zeta = \frac{1}{z} = \frac{d_{(i)}}{B_i F}. \tag{2}$$

and z, B_i, and F are the distance, baseline, and focal length. Substituting equation (2) into (1), we have the SSD with respect to the inverse distance,

$$SSD(x, y, \zeta) \equiv \sum_{k \in W} \sum_{l \in W} (f_0(x+k, y+l) - f_i(x + B_i F \zeta + k, y+l))^2. \tag{3}$$

at position (x, y) for a candidate inverse distance ζ.

First the SSD functions for individual stereo pairs are calculated in inverse distance. Then these SSD functions are simply added to produce the SSSD-in-inverse-distance such that

$$SSSD(x, y, \zeta) = \sum_{i=1}^{n} \sum_{k \in W} \sum_{l \in W} (f_0(x+k, y+l) - f_i(x + B_i F \zeta + k, y+l))^2. \tag{4}$$

The range estimate is the calculated minimum of the SSSD-in-inverse-distance curve, which shows a unique and clear minimum at the correct matching position. Though disparity changes for different baselines, there is one true distance value at each pixel. Disparity is a function of the baseline. If the SSD functions are calculated in disparity, the minimum position of each SSD function occurs at a different position. Fig. 2 shows the shapes of SSDs and SSSD functions in inverse distance for original images containing a repetitive grid pattern in the background. Fig. 3 and fig. 4 show the shapes of the SSDs of the shorter baseline pairs and longer baseline pairs respectively. The SSDs of the shorter baselines have a unique minimum

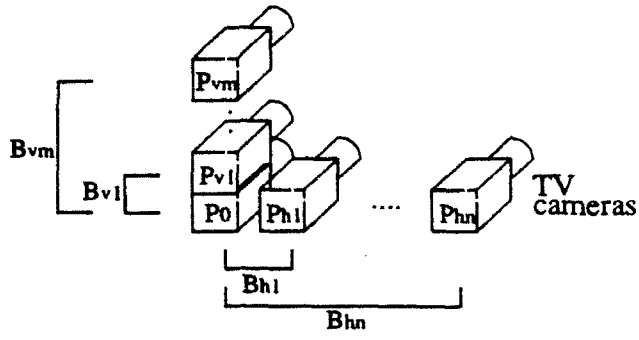


Fig. 5: Camera position for stereo in horizontal and vertical orientations

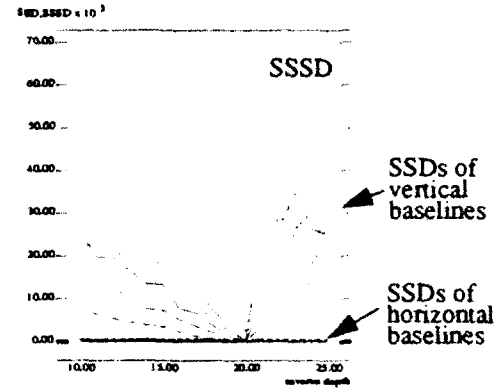


Fig. 6: SSDs and SSSD in horizontal and vertical orientations

but don't indicate a precise position, while the SSDs of the longer baselines have many minima but indicate a precise position. The SSSD, the sum of these SSDs, has a unique and precise minimum at the true position without suffering from the many minima of the longer baselines.

2.2. Use of horizontal and vertical baselines

The basic algorithms were developed to use horizontal image displacement. Some false matches, however, are caused by the orientation of the features in a scene. When the features are almost parallel to the epipolar line, we cannot obtain a good distance estimate. The solution is to use additional stereo image pairs which have epipolar lines perpendicular to the original epipolar line. Combining the information from baselinesof different orientation is straightforward as this algorithm simply adds the SSD-in-inverse-distance instead of the disparity. The effectiveness of this technique is demonstrated with an example. The camera positions for horizontal and vertical baselines are shown in fig. 5. Suppose all the features at a point are horizontal, like point "A" in fig. 37, Fig. 6 shows the shapes of the SSDs and the SSSD at this point. The SSSD shows a clear minimum. This minimum is produced by the SSDs of the vertical baselines, while the SSDs of the horizontal baselines do not have a minimum at the same position. The same applies to the vertical features, but in this case, the SSDs of the horizontal baselines mainly contribute to the minimum of the SSSD.

2.3. Detection of false matches

The shapes of the SSDs-in-inverse-distance indicate the reliability of a match and suggest the causes of false matches. We will examine the shapes of the SSD and the SSSD in three typical cases: a good match, a false match with an occlusion, and a false match due to sparse features.

Fig. 7 plots 12 curves of individual SSDs and the resultant SSSD for a point whose depth is precisely and accurately estimated like a point "A" on the sand in fig. 34. We observe that the minimum of the SSD of each baseline takes place at the same position and the curvature of the SSD near the minimum of the SSSD becomes sharper as the baseline becomes longer. The SSSD exhibits a unique and clear minimum at the correct matching position.

Let us approximate individual SSD's curves by a quadratic equation near the minimum position. From Equations (22) - (29) in [OK91], the following is expected:

- The inverse depth at which the SSD values take the minimum is expected to be the same over the various baselines.
- The curvature at which the SSD values take the minimum is proportional to the square of the baseline length.

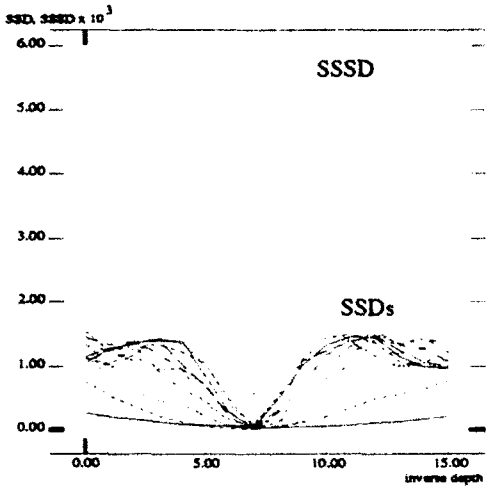


Fig. 7: SSD and SSSD values of a good match

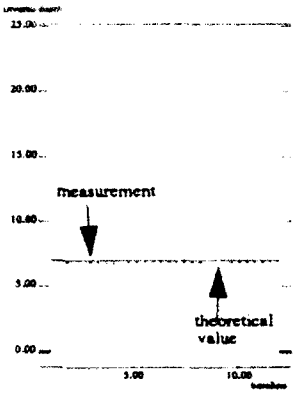


Fig. 8: Inverse depth of a good match over baselines

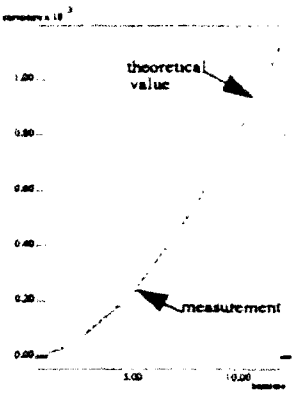


Fig. 9: Curvature of a good match over baselines

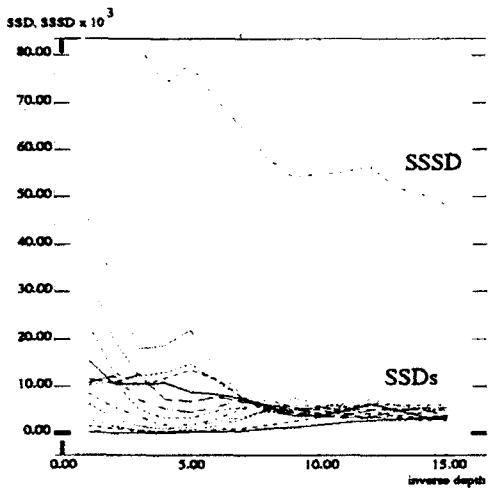


Fig. 10: SSD and SSSD values of a false match with an occlusion

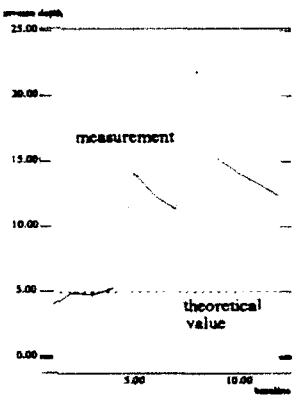


Fig. 11: Inverse depth of a false match with an occlusion over baselines

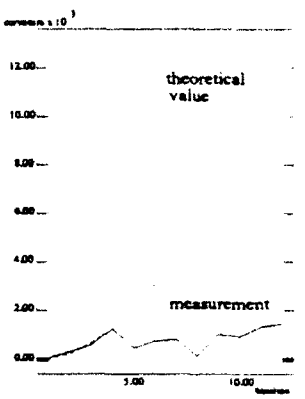


Fig. 12: Curvature of a false match with an occlusion over baselines

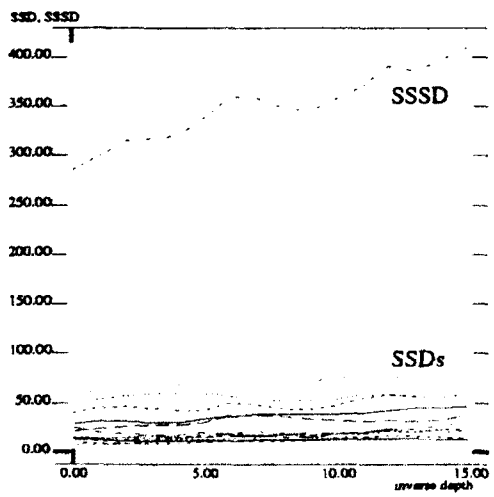


Fig. 13: SSD and SSSD values of a false match with sparse features

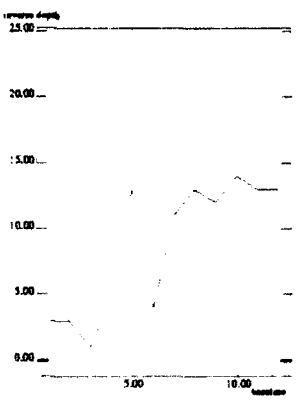


Fig. 14: Inverse depth of a false match with sparse features over baselines

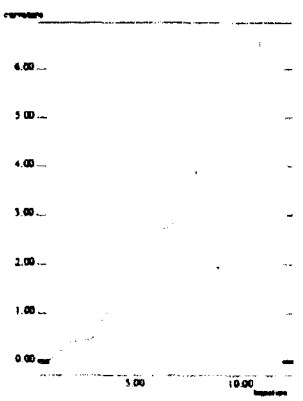


Fig. 15: Curvature of a false match with sparse features over baselines

- Variance of the final estimated inverse depth is inversely proportional to the square of the baseline length.

Fig. 8 and fig. 9 show the theoretically expected values and experimental measurements for the case of a good match as shown in fig. 7. The measurements are in good agreement with the theoretical values.

In the case of occluded features, occlusion occurs at a point which is seen by at least one camera, but due to some obstruction, is not visible from other cameras. Typically, in the case of a partial occlusion, correspondence points exist for shorter baselines, but as the baseline becomes longer, matching becomes impossiblet. The SSD and the SSSD for the former case are shown in fig. 10 like a point "C" at the right of the first parking meter head in fig. 32. The inverse depth at the minimum of the SSD of each baseline gradually shifts from the true position to a false position. The SSSD does not show a clear minimum. As shown in fig. 11 and fig. 12, the theoretically expected values and the measurements agree where the baselines are short but differ greatly where the baselines are long. Sometimes, mismatches are caused by more severe occlusions in which matches of even the short baselines cannot be relied upon.

The third case is a point with sparse features like a point "B" at the black wall in fig. 34. As shown in fig. 13, the SSD curve of each baseline is almost flat over the inverse depth range without obvious minimum. Consequently, the SSSD does not have a clear minimum. Fig. 14 and fig. 15 show the curves for this instance. The theoretically expected values are not plotted in these figures, because these values cannot be calculated in this case. From fig. 9 and fig. 15, we observe that the curvatures is much less than for a good match.

From the above observations we can see that false matches can be detected by analyzing the shapes of the SSDs and the SSSD curves. Three parameters are used to detect and classify false matches as explained in [NK92]. The first and the second parameters are a fitting error value and the inclination of a line fitted to the minimum position data of each SSD. The last parameter is the maximum value of the curvatures of all baselines. False matches are classified into three types; type O, type S, and type X. Type O is caused by an occlusion with some stereo pairs and is detected when the fitting error and the maximum curvature are large. Type S is caused by sparse features and is detected when the fitting error is large and the maximum curvature is small. Type X is caused by an occlusion with all stereo pairs or other causes of false matches and is detected when the fitting error is small and the inclination of the fitted linear line is large.

2.4. Correction of an estimate

In the previous subsection, false matches are classified into three types: type O, type S, and type X. This section shows a method to correct false matches of type O or X. For type S, no estimate is produced since there are not enough features for matching in this case.

In the case of a type O match, the SSDs of shorter baselines are chosen to calculate the estimate since occlusion occurs in longer baselines as shown in fig. 10. We choose SSDs from the shortest baseline as long as the chosen SSDs satisfy two conditions. First, that the difference between the minimum position at that baseline and that of the shortest baseline is small.Secondly, that the curvature increases as the baseline becomes longer.

Type X matches are caused by an occlusion affecting all stereo pairs or by other unusual situations. In the former case, the minimum positions of all SSDs are incorrect. We cannot correct an estimate. The latter case includes several causes of false matches, however, small curvatures of the shorter baseline SSDs are most common. In this case, we can choose the SSDs from the longest baselines to calculate our estimate as long as the chosen SSDs satisfy two conditions: the difference between the minimum position of the SSD that of the longest baseline is small, and the curvature decreases as the baseline becomes shorter.

3. Programs

Three programs have been developed to test multiple-baseline stereo:

mb-h: a basic program for horizontal baselines

mb-hd: a program for horizontal baselines with detection and correction of false matches

mb-hv: a program for horizontal and vertical baselines with detection and correction of false matches.

Subsection 3.1. describes the programs for horizontal baselines. Subsection 3.2. explains the program for horizontal and vertical baselines. The usage of these programs is described in subsection 3.3.

In our experiments, the disparity for the stereo pair with the longest baseline B_n was used. We normalize the disparity values of individual stereo pairs to the corresponding values for the largest baseline as shown in appendix C in [OK90]. We introduce the baseline ratio Ri such that

$$R_i = \frac{B_i}{B_n} \tag{5}$$

Then,

$$B_i F\zeta = R_i B_n F\zeta = R_i d_{(n)}. \tag{6}$$

Where $B_1, B_2, \ldots, B_n$ are baselines for each stereo pair and $d_{(n)}$ is the disparity for the stereo pair with baseline B_n. Substituting this into equation (4),

$$SSSD(x, y, d_{(n)}) = \sum_{i=1}^{n} \sum_{k \in W} \sum_{l \in W} (f_0(x+k, y+l) - f_i(x + d_{(n)} + k, y+l))^2. \tag{7}$$

There are three differences between Okutomi and Kanade's program [OK91] and the programs described in this paper. The first difference is that, for speed, we used single precision (32 bit) floating point, while double precision (64 bit) was used in [OK91]. The second difference is in the interpolation method used on the input intensity images. We use a linear interpolation method, while Okutomi used a cubic spline interpolation method. These first two changes, made for speed reasons, had little impact on the results we obtained. The third change was in the method used to calculate sub-pixel resolution disparities. We use a quadratic equation fitting at the position of the minimum of the SSSD, while Okutomi used an iteration method for the stereo pairs with the longest baseline. This makes a significant difference in calculation time and quality of the estimates. The results of these changes are that our programs produce results of slightly lower quality, but in greatly reduced time.

3.1. Programs for horizontal baselines

There are two programs for horizontal baselines, "mb-h" and "mb-hd." "Mb-h" is a basic program. "Mb-hd" has the capability to detect and to correct false matches. First, "mb-h" is explained. Next, the detection and correction parts of "mb-hd" are explained.

"Mb-h" uses the right-most image as a reference image, and is composed of four parts: reading input images, finding the disparity search range, calculating a pixel resolution disparity, and computing a sub-pixel resolution disparity as shown in fig. 16. The first part reads input images and keeps them in memory. The next part finds the disparity search range, calculates the SSDs of shorter baselines and determines the search range. The third part calculates the SSD of each baseline, and finds the pixel-resolution position of the minimum of the SSSD in the search range. The last part computes the sub-pixel-resolution position of the minimum of the SSSD by fitting a quadratic curve at the minimum of the SSSD detected in the previ-

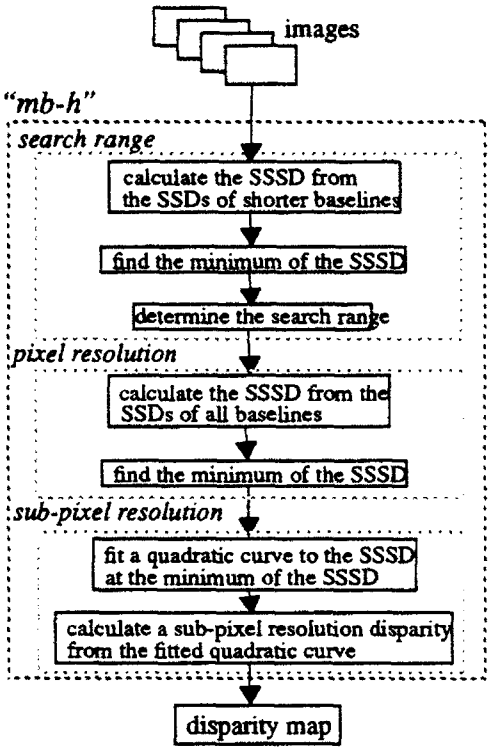


Fig. 16: Flow chart of "mb-h"

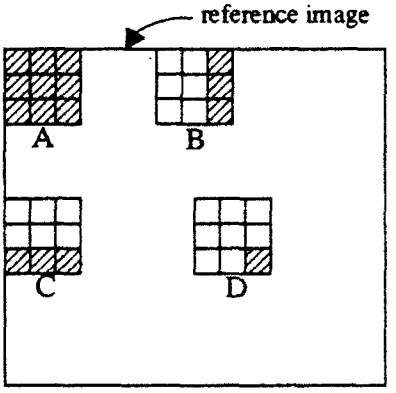


Fig. 17: SSD calculation

ous step.

Depending on the location of the window in the reference image, one of four different methods (as shown in Fig. 17) is chosen to compute the SSD. The initial window in the upper left-hand corner must be computed from scratch. Since sums of window columns are maintained however, subsequent windows can be calculated with much less effort. As shown by the shaded areas in Fig. 17, only the newly added pixels must be added in to compute a new window.

"Mb-hd" is almost the same as "mb-h" except that it can detect and correct false matches. This occurs after the calculation of the pixel resolution disparity and before the calculation of the sub-pixel-resolution disparity. A flow chart of this processing is shown in fig. 18. First, the program finds the minimum for each SSD curve. Next, the program fits a quadratic equation to the minimum position of each SSD to obtain a sub-pixel resolution disparity. The maximum curvature parameter is obtained at this step. Next, a straight line is fitted to the baseline vs. inverse depth curve. A fitting error parameter and an inclination of the fitted line parameter are obtained. Then, using these parameters, matches are classified into four categories, a good match, or a bad match of type O, type S, or type X. The final computation is the correction of false matches. When a match is classified as type O or type X, the SSSD is re-calculated with reliable SSDs and then the program again finds the minimum position of the SSSD in pixel resolution. Final output of this program is a disparity map, a variance map, and a false match map. The false match map shows the detected false matches and their types.

Fig. 19 is a detailed flow chart showing the method used to classify matches. First the fitting error value is examined. If the fitting error is larger than a threshold, the match is classified as type O or type S. A large maximum curvature indicates a type O error, while a small maximum curvature indicates an error of type S. If the fitting error is small, the slope of a straight line fitted to the curve is examined. . If the slope is larger than a threshold, the match is classified as type X. If this slope is smaller than our threshold the sign of the maximum curvature is checked. If this is positive, the match is a good match. If the sign is negative, the maximum curvature value is checked again. A small value indicates an error of type S, while a

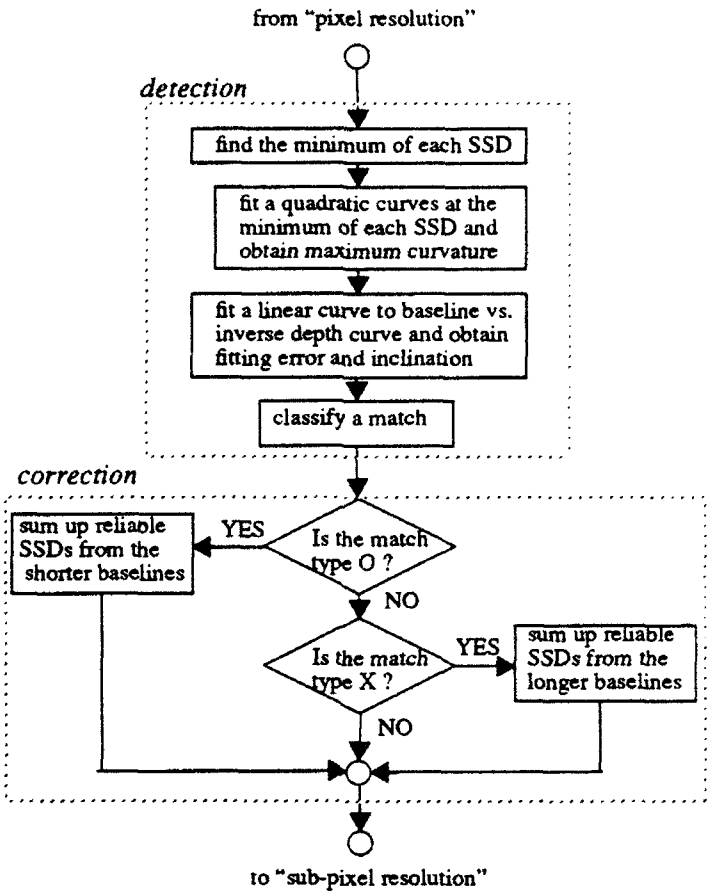


Fig. 18: Flow chart of detection and correction

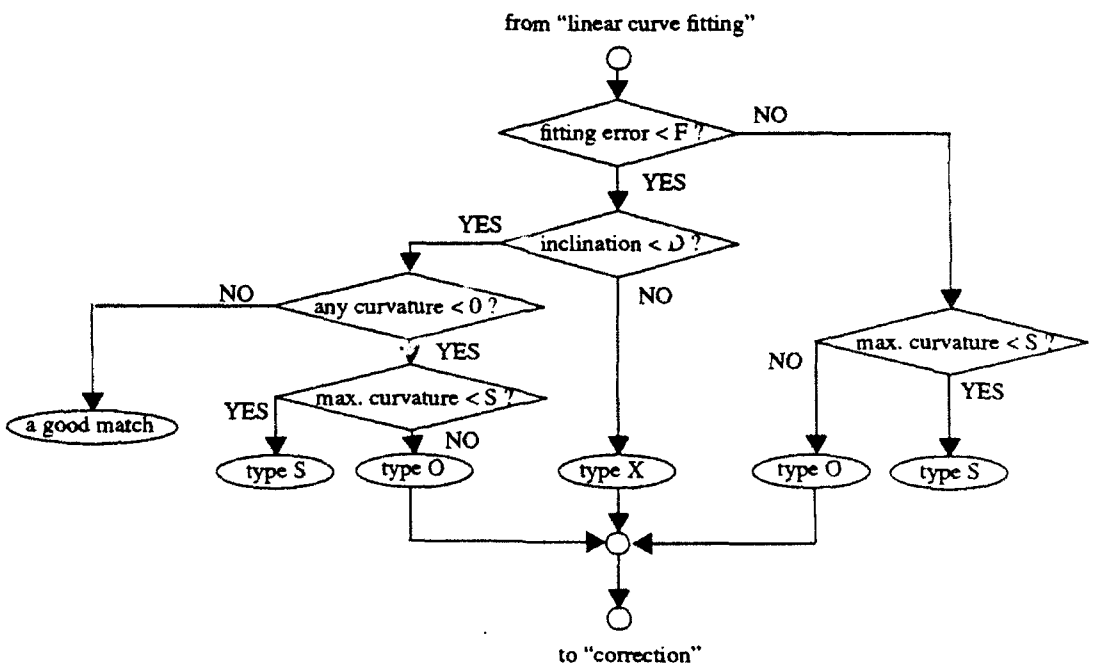


Fig. 19: Flow chart of classification

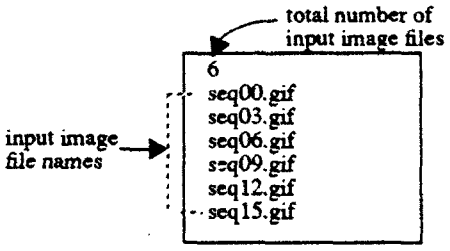


Fig. 20: Image sequence file

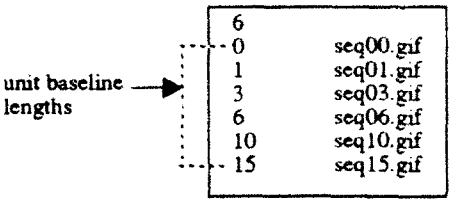


Fig. 21: Image sequence file for an arbitrary baseline ratio mode

large value indicates a type O error. In fig. 19, "F", "D", and "S" are threshold values for the fitting error, the slope of the fitted line, and the maximum curvature.

3.2. A program for horizontal and vertical baselines

A program for horizontal and vertical baselines is "mb-hv." This program is an expansion of "mb-hd " The expanded parts of "mb-hd" are part multiplying of an aspect ratio to images of vertical baselines and part the classification of a match. The aspect ratio problem is caused by the different sampling rates in row direction and in column direction on a frame memory board. We obtain the aspect ratio as a ratio of a focal length in vertical orientation and a focal length in horizontal orientation. The part of the classification of a match of "mb-hv" is a little different from the one of "mb-hd." The horizontal baselines and the vertical baselines produce their own classification results. This program uses a classification result whose variance is smaller than the another direction's variance, because the result with a small variance is reliable.

3.3. Usage of the programs

The usage of each of the three programs, "mb-h", "mb-hd", and "mb-hv" is explained below along with an example for each. Any of these programs can be run without arguments to produce a help listing.

3.3.1. "mb-h"

The following command line is an example for the "Coal" data set.

mb-h -w 7 -s 1.0 -n 30 -x 40 h6.seq h6.d.gif h6.v.gif h6.j.gif

where options; "-w", "-s", "-n", "-x", and files; "h6.seq", "h6.d.gif", "h6.v.gif", "h6.j.gif" are: the size of the window, the standard deviation of noise, the minimum disparity, the maximum disparity, an image sequence file, a filename for the disparity result, a filename for the resulting variance of the disparity, and a filename for the match classification file. The image sequence file is a text file containing an integer indicating the number of input images, and the names of each of the input images as shown in fig. 20. The order of the entries in this file is significant: the filenames start with the reference image and proceed along the baselline from closest to farthest images.

The "-t" option can be used to examine the shapes of the SSD and SSSD curves. When this flag is used, three data files are created: "ssd.data", "intplt_disp.data", and "intplt_crvt.data." The "ssd.data" file includes the SSD values for each baseline and the SSSD values. The "intplt_disp.data" and the "intplt_-crvtdata" files are the estimated disparity of each baseline and the curvature at the minimum position of each SSD.

Another option flag is "-r." This flag allows use of an arbitrary spacing between adjacent camera positions. When this option flag is used, the image sequence file has to be modified. Unit baseline lengths are included in front of each image file name as shown in fig. 21.

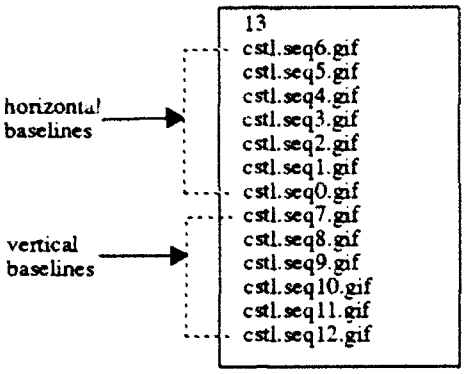


Fig. 22: Image sequence file for horizontal and vertical baselines

3.3.2. "mb-hd"

The usage of "mb-hd" is almost the same as for "mb-h" except for five additional option flags, "-l", "-c", "-F", "-D", and "-S." An example command line for the "Parking meters" data set is:

mb-hd -c -F 0.5 -D 0.83 -S 1.0 -w 7 -s 1.0 -n 1 -x 15 h8.seq h8.d.gif h8.v.gif h8.j.gif

Option "-c" activates detection and correction of false matches, options "-F", "-D", and "-S" are threshold values for the fitting error parameter, the inclination parameter, and the maximum curvature parameter. If "-c" is not chosen, the "mb-hd"program does the same thing as "mb-h."

A final option, "-l", is used to change the reference image from the right-most image to the left-most image.

3.3.3. "mb-hv"

The "mb-hv" program adds a new "-A" option which specifies an aspect ratio. Other options are the same for "mb-h" or "mb-hd." The following is an example for the "Castle" data set.

mb-hv -A 1.1752 -c -l -F 0.25 -D 0.5 -S 1.0 -w 7 -s 1.0 -n 22 -x 30 hv13.seq hv13.d.gif hv13.v.gif hv13.j.gif

The image sequence file is different from the ones of the previous programs for horizontal baselines. First, the input image files in the horizontal direction are listed, then the input image files in the vertical direction follow as shown in fig. 22.

4. Experimental results

Our multiple-baseline stereo system has been tested with miniature model town scenes and outdoor scenes using horizontal baselines and using combined horizontal and vertical baselines. The miniature model town sceneswere obtained under well controlled conditions in the Calibrated Imaging Laboratory. The outdoor scenes include variable lighting conditions and large distance variations. Subsections 4.1. and 4.2. present the results of experiments with the basic algorithm. Results in detecting and correcting false matches are shown in subsections 4.3. and 4.4.

4.1. Results with horizontal baselines

We have used five scenes for this experiment. Two of them are the miniature model town scenes and the others are outdoor scenes. The experimental setup for acquiring stereo pairs is illustrated in fig. 23. The

images are acquired by moving a camera horizontally. The distance between adjacent camera positions is constant. Table 1 describes the image acquisition parameters. For a typical miniature model town scene, the distance from the camera to the nearest object is 0.51m and the baseline length ranges from 1.27 mm for the closest camera pair to 11.43 mm for the farthest. For a typical outdoor scene, the distance from the camera to the nearest object is 19m and the baseline length ranges from 19.05mm to 114.3mm.

As illustrated in fig. 24, images are preprocessed with a Laplacian of Gaussian (LOG) filter to reduce photometric distortion. A 5x5 window is used for the Gaussian and a 3x3 window is used for the Laplacian. Then the multiple-baseline stereo method is used to compute the inverse depth with a 7x7 window for the SSD computation. Typically, for a miniature model town scene, the number of the stereo pairs is 9, the image size is 256x240, and the total disparity range is10 pixels. For an outdoor scene, the number of stereo pairs is 6, the image size is 240x256, and the total disparity range is 9 pixels, as summarized in table 2.

4.1.1. Coal

Fig. 25 shows the "Coal" data set which consists of five stereo pairs. The maximum disparity between the adjacent images is approximately two pixels. Fig. 26 is the LOG preprocessing result of one of the original images. Fig. 27 is the isometric plot of the resultant depth map. The depth map shows no gross errors. The shapes of the buildings are well estimated. We can see a chimney on top of the upper-right building, the slant of the roof of the upper-left building, and a flag on the tower in the center.

4.1.2. Town

The "Town" data set is shown in fig. 28. The background has a repetitive grid pattern. The isometric plot of the resultant depth map is shown in fig. 29. Note that the background is well determined. We observe the depth differences between buildings and the shapes of each tree in the center. A large error occurs at the roof in the center of the scene because of sparse features.

4.1.3. Shrubbery

Fig. 30 Shows the "Shrubbery" data set. This data set has a repetitive brick wall pattern in the background. The isometric plot of the resultant depth map is shown in fig. 31. We observe that the shrubs at the left and in the center are well separated, and the depth jump around the sign board and the top of the signpost are clearly distinct from the wall. The area of the wall is well estimated, however the depth map is a little noisy. We can see a round bush at the right. Some mismatches are observed at the curb, because the features in this area are almost parallel to the epipolar line.

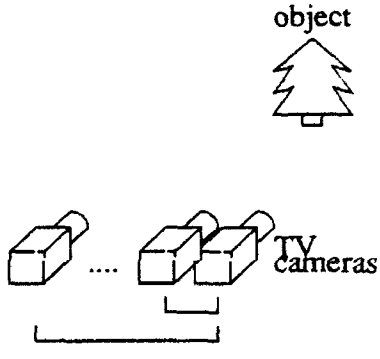


Fig. 23: Setup for horizontal baselines

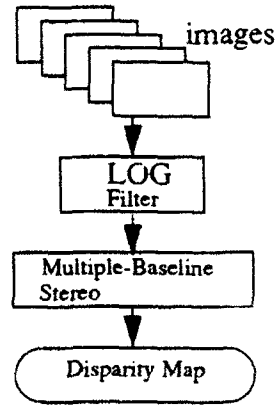


Fig. 24: Procedure

Table 1 Image acquisition parameters

Name	Distance		Baseline length		TV camera	Focal length
	the nearest	the farthest	unit	longest		
Town	0.51m	1.02m	1.27mm	11.43mm	———	———
Coal	——	——	7.62mm	38.10mm	———	———
Shrubbery	19m	28m	19.05mm	114.30mm	SONY XC57	50mm
Parking meters	12m	34m	10.16mm	71.12mm	SONY XC57	50mm
Sand	6m	10m	2.54mm	12.70mm	SONY SSC-D7	50mm
Shrubbery2	19m	28m	20.0mm	60.00mm	SONY XC57	50mm
Corner	19m	28m	20.0mm	60.00mm	SONY XC57	50mm
Guide	16m	90m	20.0mm	60.00mm	SONY XC57	50mm
Boxw	5.18m	——	15.0mm	90.00mm	SONY XC77	24mm
Charge	5.89m	9.23m	8.0mm	80.00mm	Fuji Electric	25mm
Coal2	1.52m	1.70m	5.0mm	30.00mm	SONY XC77	50mm
Train2	2.25m	2.95m	5.0mm	30.00mm	SONY XC77	50mm
Castle	1.50m	1.70m	5.0mm	30.00mm	SONY XC77	50mm
Field	——	——	15.0mm	90.00mm	SONY XC77	24mm
Hill	——	——	15.0mm	90.00mm	SONY XC77	24mm

Table 2 Image processing parameters

Name	Number of stereo pair	Image size	Disparity range	Processing * time (sec.)
Town	9	256x240	4 - 14	332
Coal	5	256x240	30 - 40	166
Shrubbery	6	240x256	4 - 13	209
Parking meters	7	240x256	1 - 15	351
Sand	5	240x256	1 - 6	102
Shrubbery2	H : 3 V : 3	240x256	1 - 7	-----
Corner	H : 6 V : 6	240x256	1 - 7	350
Guide	H : 3 V : 3	240x256	0 - 8	-----
Boxw	H : 3 V : 3	240x256	10 - 20	-----
Charge	H : 5 V : 5	240x256	8 - 15	-----
Coal2	H : 6 V : 6	240x256	27 - 35	290
Train2	H : 6 V : 6	240x256	10 - 25	607
Castle	H : 6 V : 6	240x256	22 - 30	305
Field	H : 6 V : 6	240x256	0 - 12	510
Hill	H : 6 V : 6	240x256	0 - 12	510

*Processing time was measured on a SUN 4/75 (28MIPS/4.5MFLPS) with 16MB memories.

4.1.4. Parking meters

The "Parking meters" data set includes seven stereo pairs. Fig. 33 is the isometric plot of the depth map. The following portions in the scene are well estimated: the three parking meters in front of the shrubs, the side view of the sign board which is between the second and the third parking meters, and the large depth gap between the near and far parts of the building. There are some mismatches at the back door of the car because of sparse features in this area.

4.1.5. Sand

The scene of "Sand" contains rough, natural surfaces like sand and rocks as shown in fig. 34. Five stereo pairs were taken for this data set. Fig. 35 is the isometric plot of the depth map. We observe that the two rocks and the sand are well estimated. Many mismatches, however, occur at the border between the black wall and the white curtain. The features in this portion are parallel to the epipolar line and are also somewhat sparse.

4.2. Results with horizontal and vertical baselines

We also performed experiments which used stereo image sets produced with both vertical and horizontal baselines. Fig. 36 illustrates the experimental setup. The procedure is the same as in the horizontal baselines experiment, except images are taken by moving a camera both horizontally and vertically. The acquisition parameters are shown in the last ten rows of table 1. (Some data sets are not shown in this paper). For a typical miniature model town scene, the distance from the camera to the nearest object is 2.25m and the baseline length ranges from 5 mm for the closest camera pair to 30 mm for the farthest. For a typical outdoor scene, the distance from the camera to the nearest object is 19m and the baseline length ranges from 20mm to 60mm.

The last ten rows in table 2 show the image processing parameters. Typically, for a miniature model town scene, the number of the stereo pairs is 6 for each baseline and the total disparity range is 15 pixels. For outdoor scenes, the number of the stereo pairs is 6 for each baseline and the total disparity range is 6 pixels.

Subsections 4.2.1., 4.2.2., and 4.2.3. show the results produced from three model town scenes, "Train2", "Coal2", and "Castle." The results from three outdoor scenes, "Corner", "Field", and "Hill" are shown in subsections 4.2.4. and 4.2.5

4.2.1. Train2

Fig. 37 shows the "Train2" data set which consists of twelve stereo pairs of which six have horizontal baselines, and six have vertical baselines. The maximum disparity between adjacent images is spproxi-

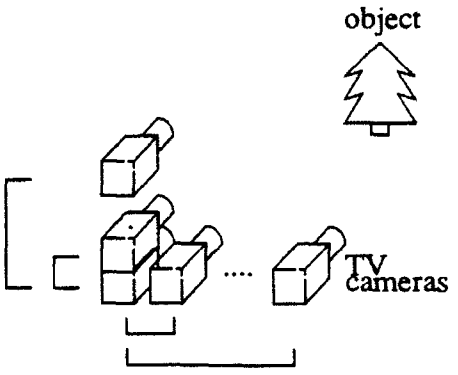


Fig. 36: Setup for horizontal and vertical baselines

mately two pixels. The isometric plot of the resulting depth map is shown in fig. 38. The shape of each building is well estimated. The corresponding oblique view of the scene is also shown at the upper right-hand corner of this figure. It is easy to recognize correspondence between the isometric plot and the oblique view.

Although many image features are almost parallel to the epipolar lines of horizontal image pairs or to those of the vertical image pairs, there are no gross matching errors in the depth map. An examination of the SSD and SSSD values shows us why. A point with horizontal image features, such as point "A" in figures 27-39, shows a poor minimum for the SSDs of the horizontal image pairs. The vertical pairs, however, show SSD curves which have a clear minumum at this position. The SSSD yields a clear, and unique minimum at the correct position. Similar results are obtained in the case of features that are vertical in orientation, but are well matched by image pairs with horizontal baselines as in the case of point "B" in figures 40-42. The result of using two perpendicular baselines is that the number of matching errors is greatly reduced, and the quality of the depth map is dramatically improved.

4.2.2. Coal2

The "Coal2" data set shown in fig. 43 is a scene of a model coal mine. The oblique view of this data set is shown in fig. 44. Fig. 45 shows the isometric plot of the resulting depth map. We observe three supports at the left side of the tower and a board on top of the tower.

4.2.3. Castle

Fig. 46 and fig. 47 show the "Castle" data set and its oblique view. Well estimated features include the spiral road around the castle, the shape of the castle, and the slope at the bottom-left corner of the input image. We can also see the two small watch towers at the bottom and the right side of the input image.

4.2.4. Corner

The "Corner" data set is shown in fig. 49. Fig. 50 shows the isometric plot of the resulting depth map. The false matches from fig. 31 are improved in this new result, however, this result is noisier. This noise is due to the fact that the baseline length for this dataset is one-half of that used in the "Shrubbery" data set.

4.2.5. Field and Hill

Figures 51 and 57 are different views of a grassy field with a line of trees in the background. Isometric plots of the depth maps for these images are shown in figures 52 and 58. The shape of the grassy hill is well described, however, there are many errors in the depth map in the area of the sky, where there are very few features to aid matching. Figures 53-56 and 59-62 show elevation profiles produced from the range data. These profiles demonstrate the ability of the stereo system to estimate the shape of the hillside.

4.3. Results in detecting false matches

This section presents results produced using our method to detect, classify, and correct matching errors. The first two examples were produced from horizontal baseline datasets, while the third result used input sets with both horizontal and vertical baselines.

Figures 63-65 show where in the input image false matches of each of the three types described in section 2.3 were detected. The parameters used in this case were: F, fitting error: 0.5 ; D, inclination: 0.83 ; and S, maximum curvature: 1.0.

The locations of the false matches are reasonable. Occlusions are detected in the left-hand bush, at the first parking meter, and at the left side of the car on the right. Sparse features are detected on the back of the car and on the building wall. False matches of type X appear in the left bush, around the three parking meters, and on the left side of the car. The type X errors on the building and on the back of the car are

caused by sparse features. Figure 66 shows the isometric plot of the depth map with good matches. This result can be compared with figure 33 to see that the removal of bad matches considerably improves our results.

The next example is the "Sand" data set. The detected false matches are shown in figures 67-69. The threshold values, F, D, and S are 0.5, 0.83, and 1.0. The false matches with type O are not detected in the scene, because this scene does not have a large depth range. The algorithm detects many false matches with type S on the black wall and the white curtain, because the features are sparse in these areas. The false matches of type X indicate an area of few features. Compare the plot of only good matches, shown in fig.70 with the original plot of all matches in fig. 35.

False matches detected in the "Train" data set are illustrated in figures 71-73. Occlusions are detected at the borders between the buildings and the background. Sparse features are detected on the roofs of the buildings at the left and in the center of the image. In this case, the false matches of type X at the edge of the buildings and on the roofs of the buildings indicate occlusions and sparse features. Figure 74 shows the result with false matches removed. This result shows no large errors.

4.4. Results with corrected estimates

The false matches which were detected in the previous section can be corrected as described above. Figure 75 shows the isometric plot of the corrected depth image of the "Parking meters" data set. The occlusion on the left side of the first parking meter and the sparse features on the back of the car are corrected. Some estimates become worse when false matches are of type X, such as in the bush on the left side of the second parking meter. The correction method for the false matches of type X does not handle all false matches of this type as these errors may be caused by a variety of reasons.

Corrected results of the "Sand" data set are shown in fig. 76. The estimates at the positions of detected false matches are much improved, however the corrected estimates at the white curtain are noisy. The noisy estimates at the white curtain are caused by a lack of features in this area.

Figure 77 shows results, with corrections, of the "Train2" data set. The good quality of the original estimates in figure 38 leaves little room for improvement. Currently, it is hard to tell the difference between the original result and the result with corrected matches.

5. Data acquisition

5.1. Image acquisition

Currently, stereo pairs are produced using a single camera which is moved between images to produce a set of stereo pairs. The camera is installed on a precise X-Z table which produces coplanar camera movement. Three parameters, the number of stereo pairs, the focal length, and the total baseline length are determined at the time of imaging. We have empirically determined that at least three and, preferably five, stereo pairs should be taken to get the best results. The focal length is set according to the size of the region of interest and the distance from the camera to the objects. We prefer to use 50mm or 24mm lenses to reduce the lens distortion. Though the total baseline length is determined by the number of the stereo pairs and the unit baseline length, we find that the total baseline length should be set such that there are between 10 and 20 pixels of disparity in a 240x256 size image.

In stereo matching, the relationship between the distance, Z and the disparity, D is given as:

$$Z = \frac{F \times B}{D}. \tag{8}$$

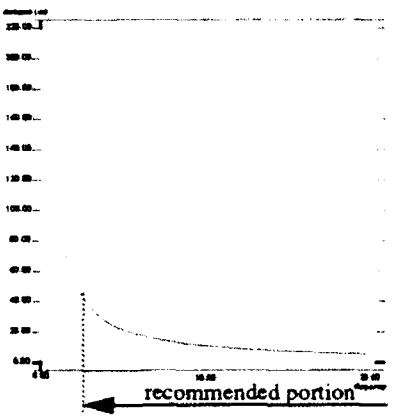


Fig. 78: Disparity vs. distance

where F and B are the focal length and the baseline length. Once the focal length and the total baseline length are determined, the disparity vs. the distance curve is plotted as shown in fig. 78. The gently sloped portion of the curve produces good distance estimates, while the steeper portion of the curve produces noisy results. If the area of interest falls in this steeper portion, the focal length or the total baseline length should be changed.

5.2. Calibration

We use a calibration method developed by Arakawa [A92]. Seven extrinsic parameters and four intrinsic parameters are calculated. The extrinsic parameters are an angle between the horizontal axis of the jig and the y axis of the world coordinate system, a translation between the world coordinate system and the camera coordinate system, and a rotation between the world coordinate system and the camera coordinate system. The translation and the rotation are composed of three components. The intrinsic parameters are the focal lengths in horizontal and vertical orientations and the image center in the image coordinate system.

The calibration procedure has three steps. The first step is acquiring a rough estimate of parameters using a linear system which is determined by the world coordinates of the targets and their projections on the images. The next is the re-computation of the rotation matrix to guarantee orthonormality. The last step is the optimization of the parameters by iterative minimization of the error of the target projection. For the first two steps, Weng's method [WCH90] is applied. The third step uses Powell's method [PFTV88]. The procedure of image acquisition for calibration and the usage of the programs for the calibration are described in the appendix.

6. Conclusion

This paper presented experimental results produced with the multiple-baseline stereo system. The algorithm was applied to miniature model town scenes and outdoor scenes. The miniature model town scenes were taken under well controlled conditions in the Calibrated Imaging Laboratory. First we used stereo pairs acquired by moving a camera horizontally and showed that this algorithm worked well with the outdoor scenes as well as the miniature model town scenes. Next we used stereo pairs taken by moving a camera in both horizontal and vertical directions. We showed that the use of the stereo pairs with two orthogonal baseline orientations removed ambiguity and increased precision without problems with the orientation of features in a scene.

We also demonstrated that the shapes of the sum of squared-difference (SSD) values near the estimate could predict the reliability of the match. Using these SSDs, the matches were classified into four categories, a good match and three false match types. The false match types are type O; occlusion, type S; sparse

features, and type X; occlusion and other false matches. The algorithm detected the false matches fairly well, and reasonably classified the false matches. We showed that the removal of poor matches greatly improved the estimates.

Finally, we demonstrated that the parameters of the classification of the match indicated a method to improve poor estimates of type O or type X. False matches were easily corrected except for false matches of type X. The correction method for false matches of type X does not cover all the false matches in this category. The type X category should be sub-divided into further match types.

We also explained the processing programs and the data acquisition process including camera calibration.

Acknowledgments

The authors would like to thank Kenichi Arakawa for developing the calibration method for this algorithm, Hans Thomas for assisting in image acquisition, Dr. Robert A. Mathias for improving the readability of the paper, and Bill Ross for his extensive editing and technical assistance.

References

[OK91] Masatoshi Okutomi and Takeo Kanade. A multiple-baseline stereo. In proc. of CVPR '91, pages 63-69, Lahaina, Maui, Hawaii, June 1991.

[NK92] Tomoharu Nakahara and Takeo Kanade. Detection of occlusions and sparse features by the multiple-baseline stereo. In preparation, 1992.

[OK90] Masatoshi Okutomi and Takeo Kanade. A multiple-baseline stereo. Technical Report CMU-CS-90-189, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, 1990.

[A92] Kenichi Arakawa. Camera calibration for multiple-baseline stereo. Internal memo, the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, 1992.

[WCH90] Juyang Weng, Paul Cohen, and Marc Herniou. Calibration of stereo cameras using a non-linear distortion model. In proc. of 10th ICPR, pages 246-253, June 1990.

[[PFTV88] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. Numerical recipes in C: The art of scientific computing. The Press Syndicate of the University of Cambridge, Cambridge, GB, 1988.

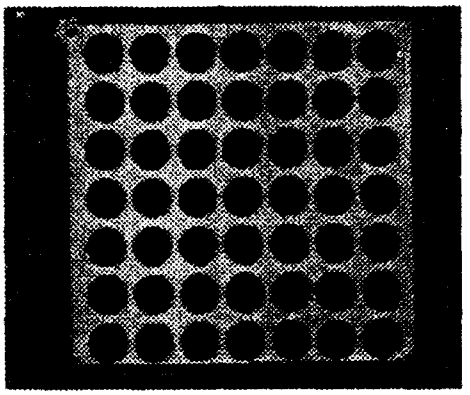


Fig. 79: Target board

Appendix

A.1 Image acquisition for calibration

The three dimensional coordinates of the targets in the world coordinate system are used as the ground truth. The measurements of the positions and the orientations of the targets and the camera are required to be accurate. The target marks are drawn on a target board. Our target board has fourty-nine circle marks. Each mark has 25mm diameter and is located on a grid point every 50mm in horizontal and vertical orientations as shown in fig. 79. Two images are taken at the different distance from the camera to the target board, because this calibration method requires that all the targets are not on a plane.

A.2 Usage of programs for calibration

This calibration method uses two programs, "DetectTargets" and "calibFromFile." The former program detects the positions of the targets in the three dimensional space and prepares the coordinates of the targets for calibration. The latter program calibrates the eleven parameters using the coordinates of the targets. The following command line is a typical example for the first program.

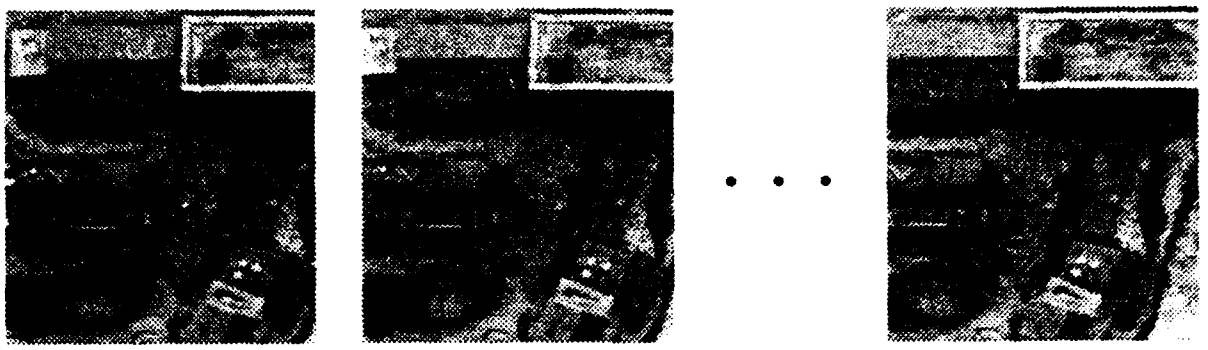
detectTarget -d xwindows -b 100 -s 150 -D 33 27 -v 0.015 -V 3 -R 75 9 75 393 416 393 416 9 img-Seqname target3DFile outputFile

where options; "-d", "-b", "-s", "-D", "-v", "-V", and "-R" are showing the results on a display, a threshold for the binarization, a threshold of the minimum target size, search ranges in vertical and horizontal orientations, a camera interval length, the number of images in horizontal orientation, and indication of moving a camera from right to left respectively. The eight numbers before the last three names are the coordinates of the four corners of the interest rectangular region on the first image. The order of the coordinates is clockwise from the top-left corner. The last three names are an image sequence name, a file name of the three dimensional coordinates of the targets, and a file name of the result. When the "-d" option is used, a xwindow have to be opened in advance. Other options are "-m" and "-g." They are a threshold of the moment of the target and a switch for drawing a grid when the "-d" is selected.

After the "detectTarget" is applied to the two images, the resulting files are edited into one file. The next command line is a typical example for the calibration program.

calibFromFile -I outputFile

where the "-I" option is for the iterative optimization of Powell's method and the "outputFile" is the edited file. The resulting eleven parameters are shown on a display.



1st (left most) 2nd 6th (right most)

Fig. 25: "Coal" data set

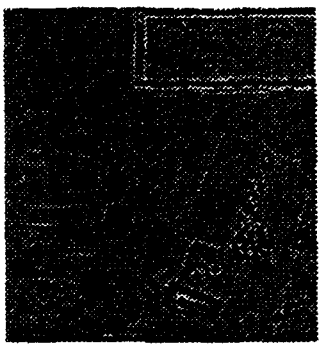
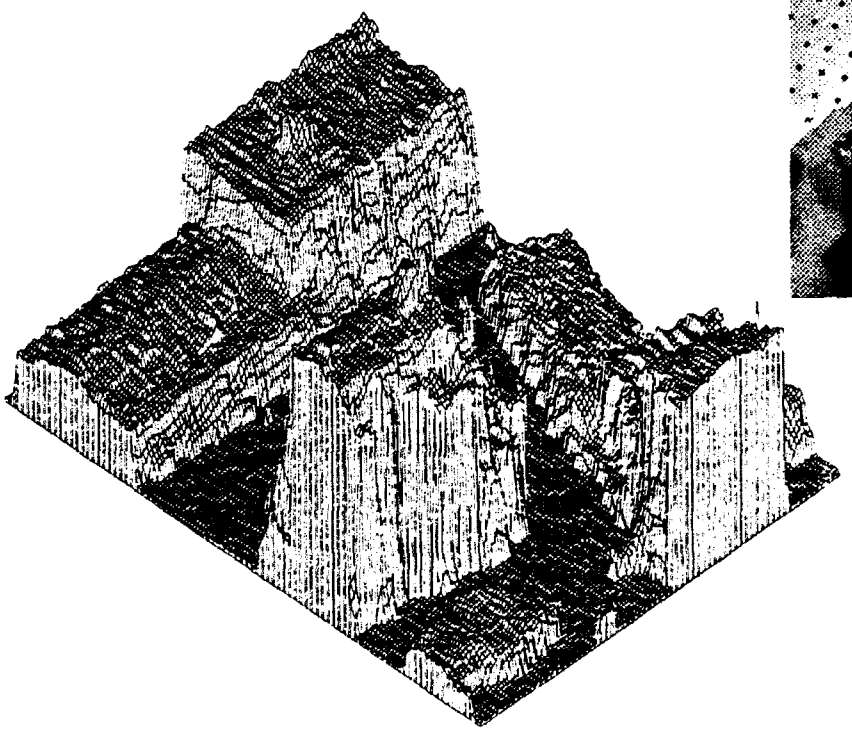
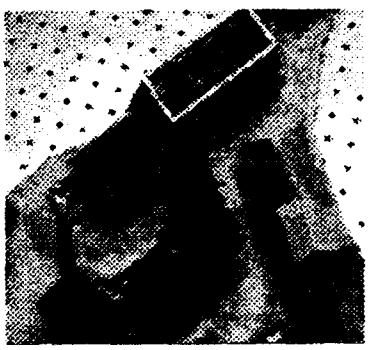


Fig. 26: Laplacian of Gaussian image





Oblique view

Fig. 27: Isometric plot of depth ("Coal")

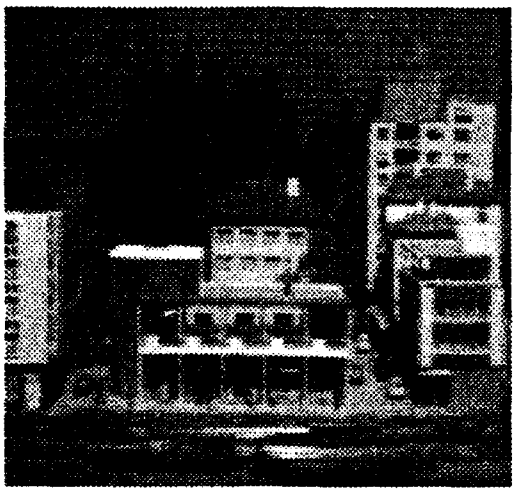


Fig. 28: "Town" data

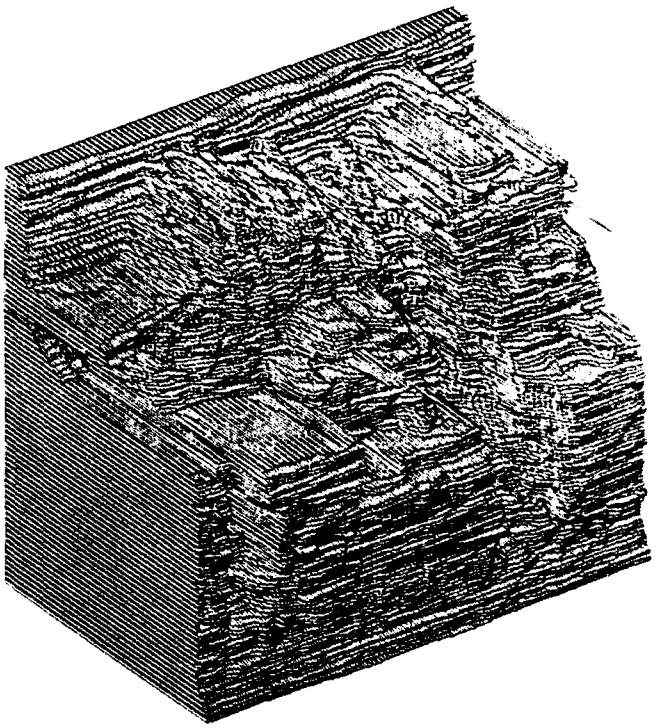


Fig. 29: Isometric plot of depth ("Town")

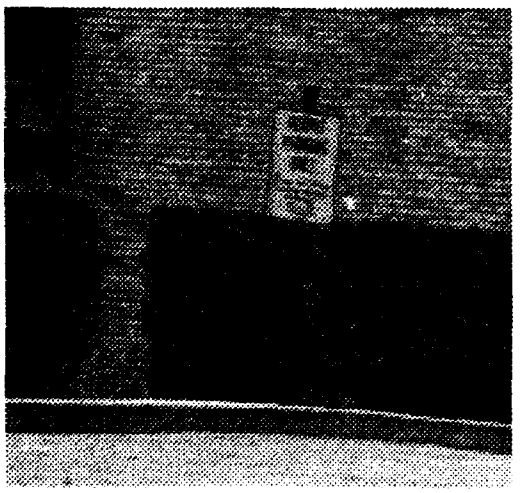


Fig. 30: "Shrubbery" data

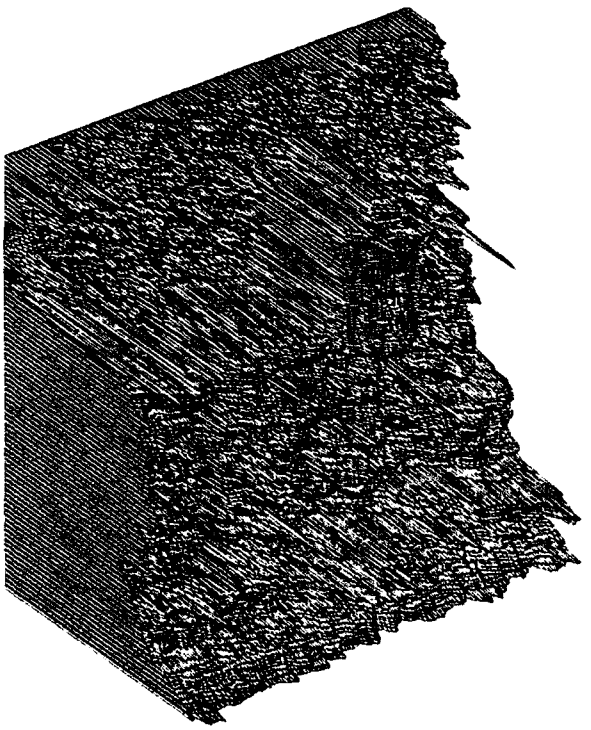


Fig. 31: Isometric plot of depth ("Shrubbery")

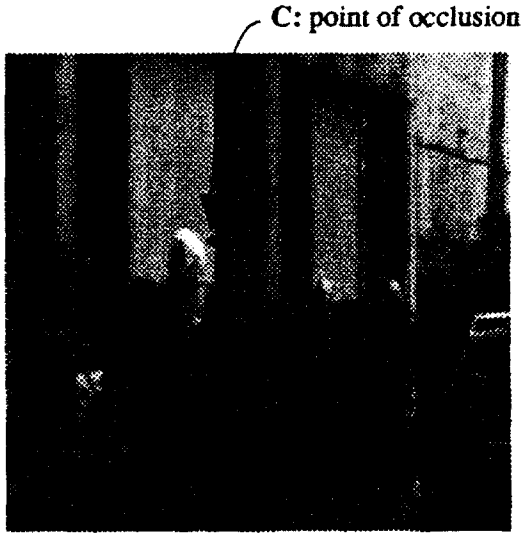


Fig. 32: "Parking meters" data

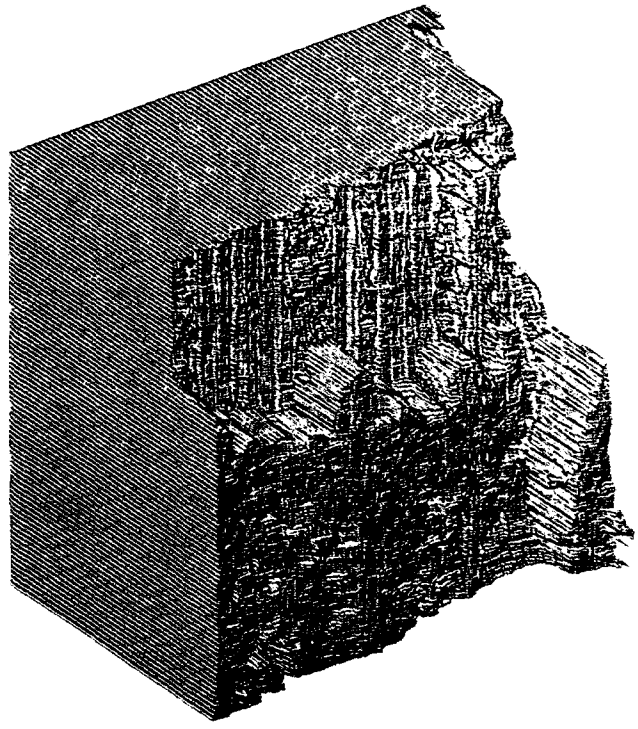


Fig. 33: Isometric plot of depth ("Parking meters")

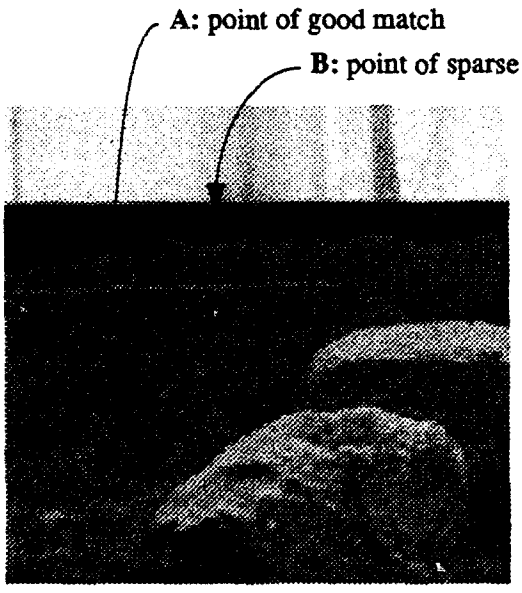


Fig. 34: "Sand" data

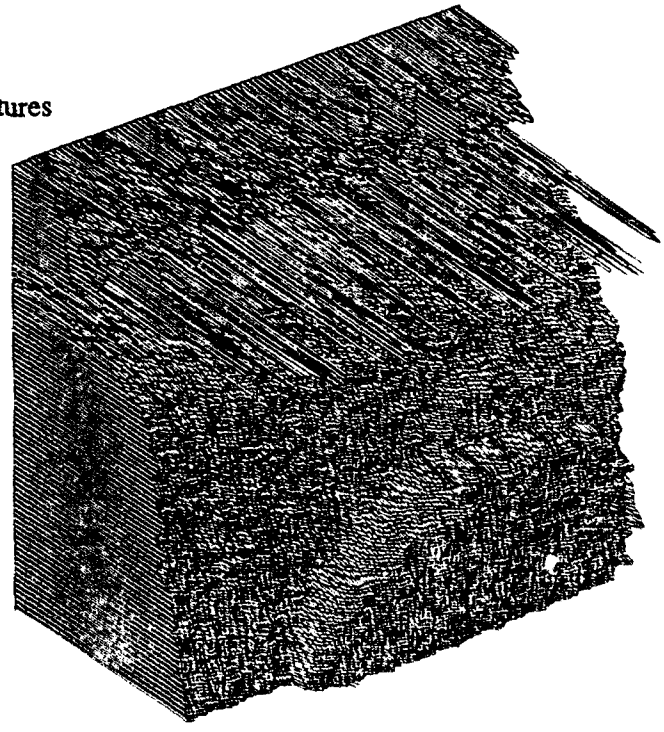


Fig. 35: Isometric plot of depth ("Sand")

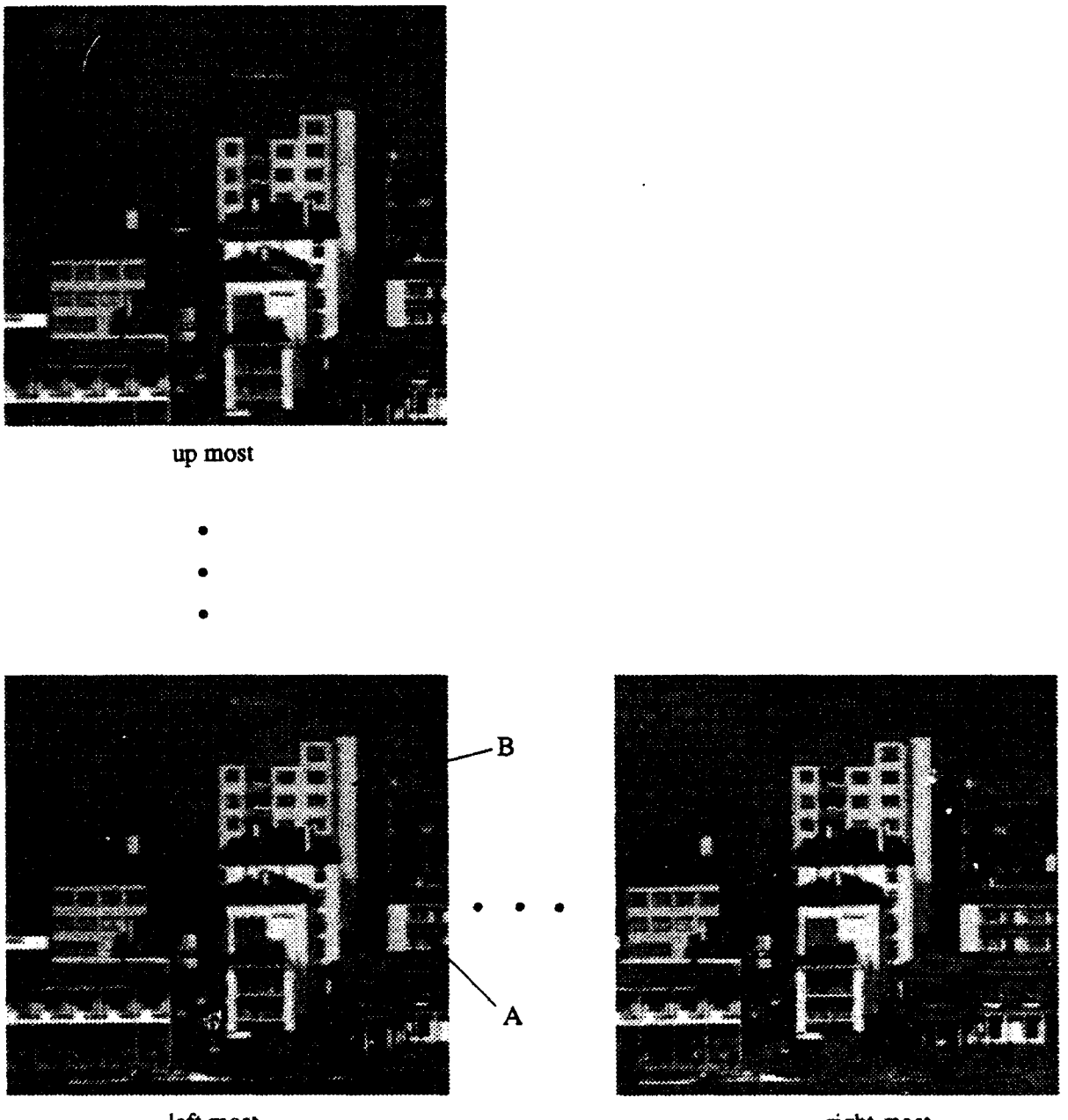


Fig.37: "Train2" data set

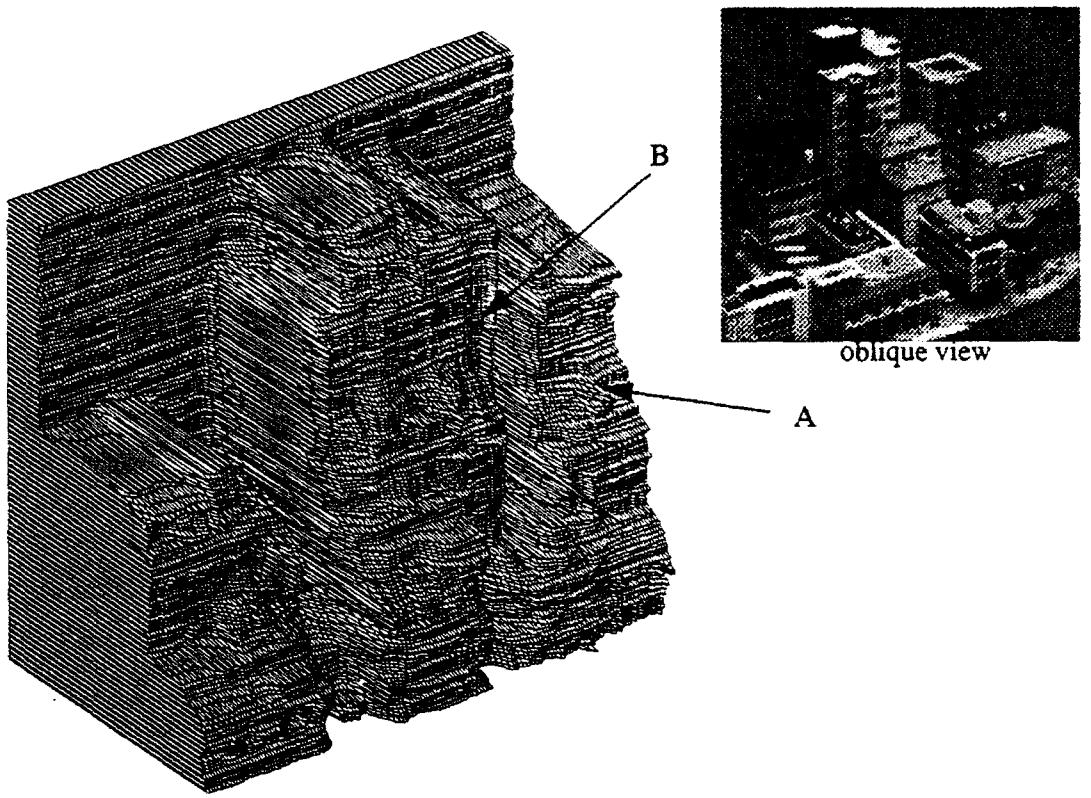


Fig. 38: Isometric plot of depth resulted from horizontal and vertical baselines ("Train2")

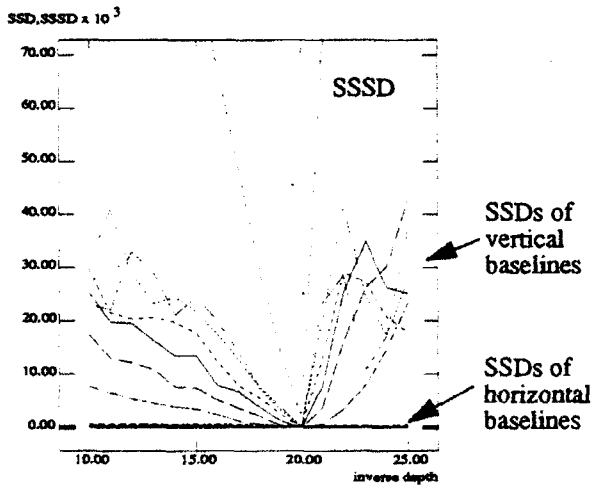


Fig. 39: SSD and SSSD values vs. inverse depth at a point "A" of a horizontal feature

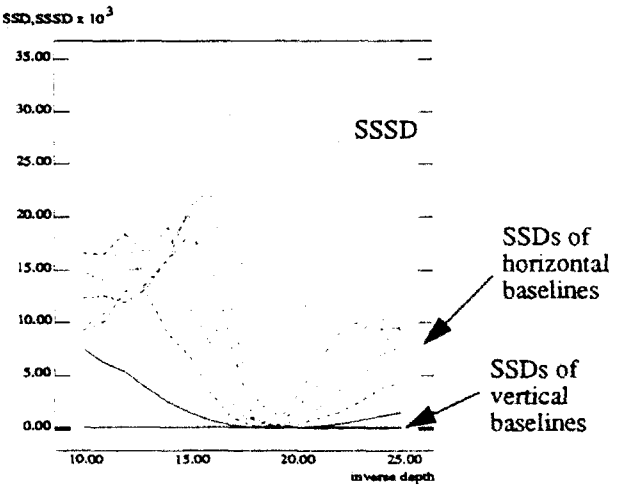


Fig. 40: SSD and SSSD values vs. inverse depth at a point "B" of a vertical feature

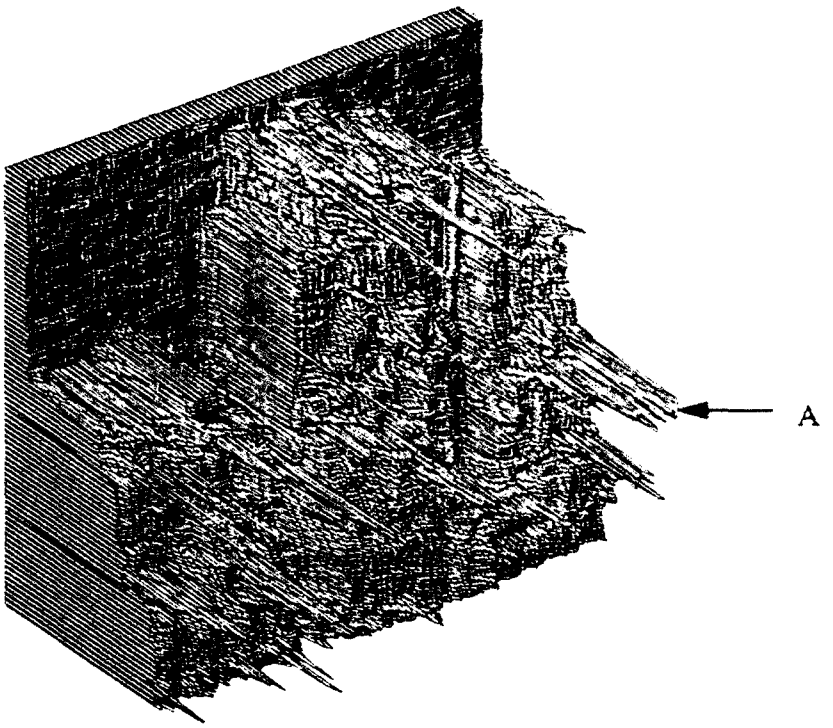


Fig. 41: Isometric plot of depth resulted from horizontal baselines ("Train2")

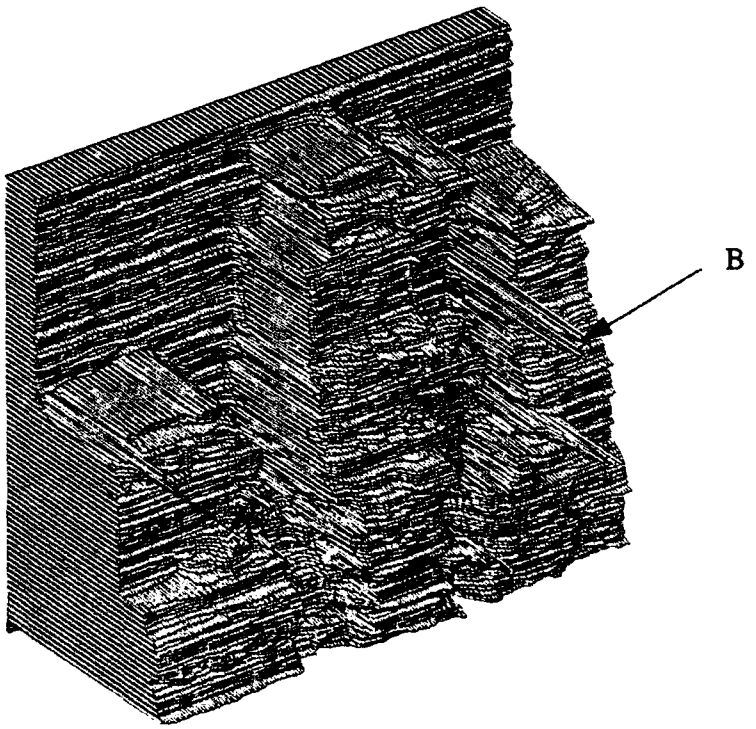


Fig. 42: Isometric plot of depth resulted from vertical baselines ("Train2")

Fig. 43: "Coal2" data

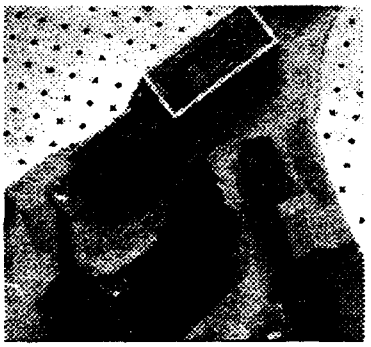


Fig. 44: Oblique view of "Coal2"

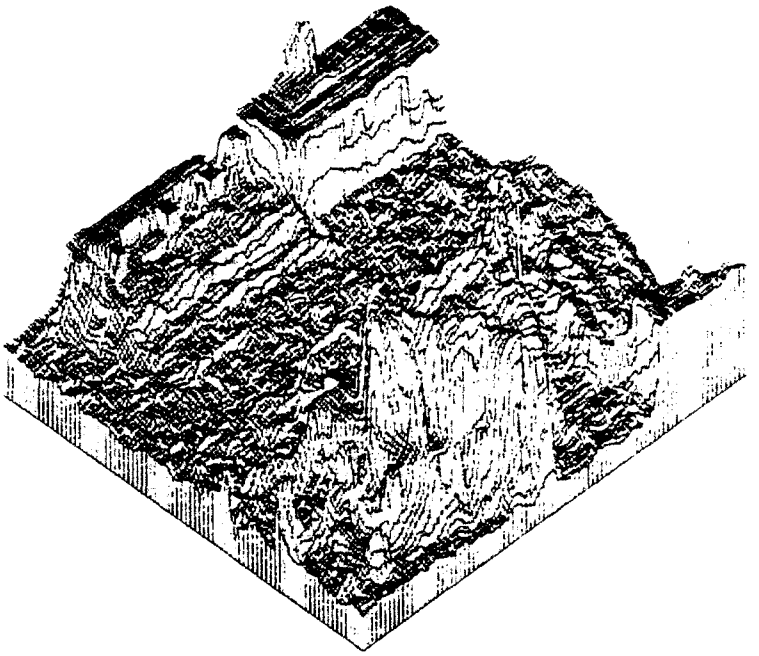


Fig. 45: Isometric plot of depth ("Coal2")

Fig. 46: "Castle" data

Fig. 47: Oblique view of "Castle"

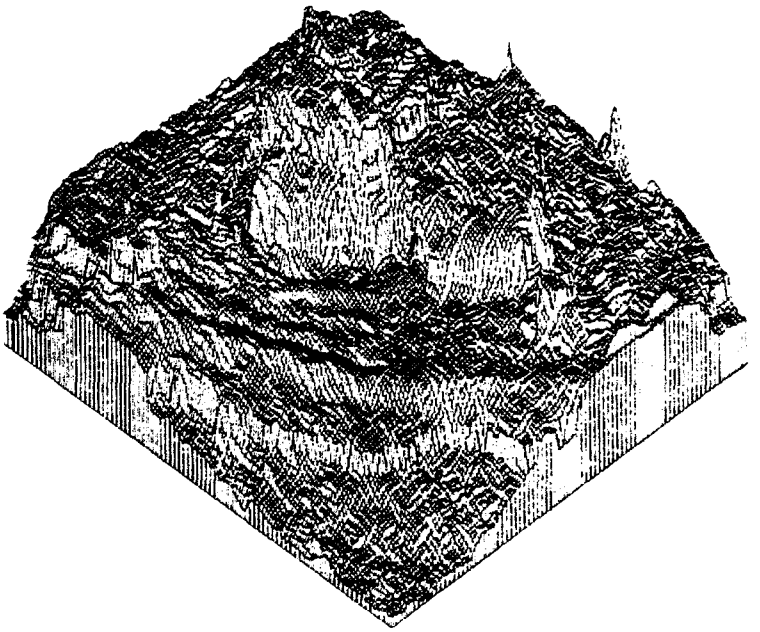


Fig. 48: Isometric plot of depth ("Castle")

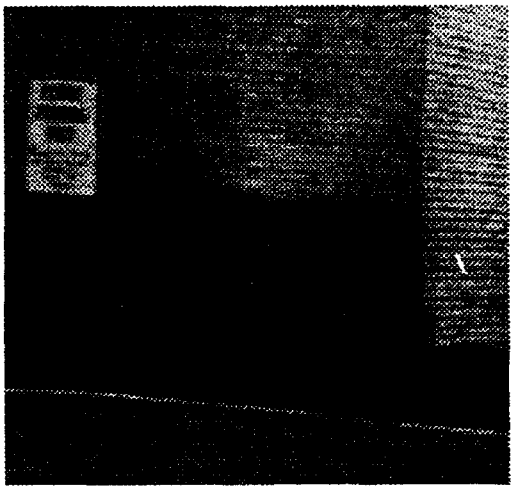


Fig. 49: "Corner" data

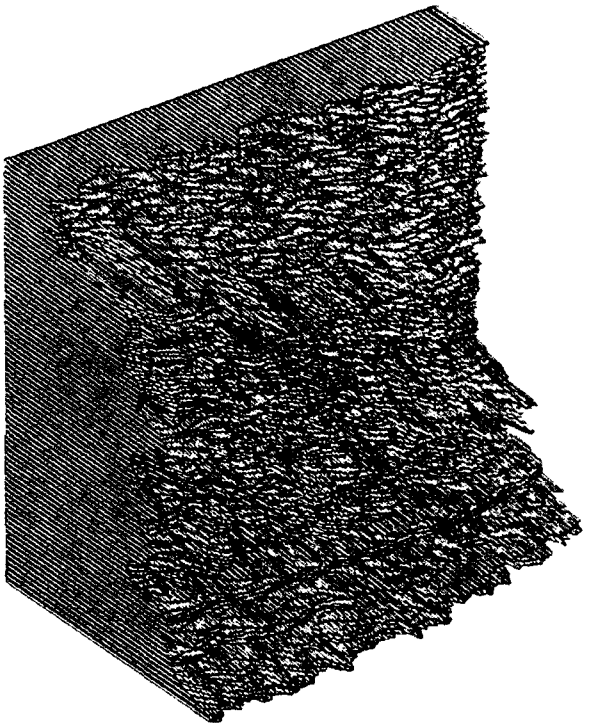


Fig. 50: Isometric plot of depth ("Corner")

Fig. 51: "Field" data

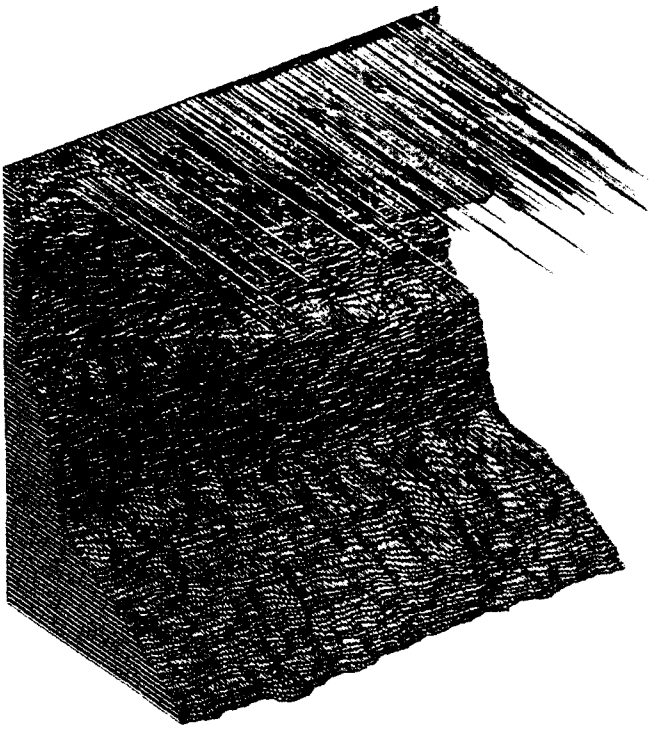


Fig. 52: Isometric plot of disparity ("Field")

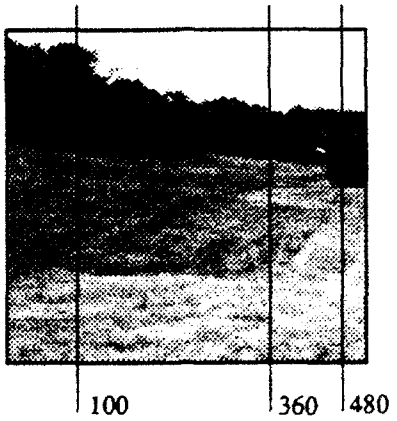


Fig. 53: Locations of profile of "Field" data set

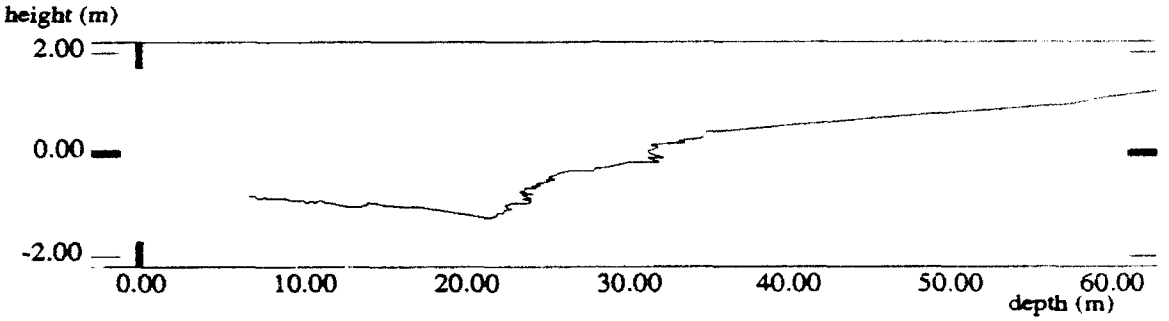


Fig. 54: Profile of depth at column 100 of "Field" data set

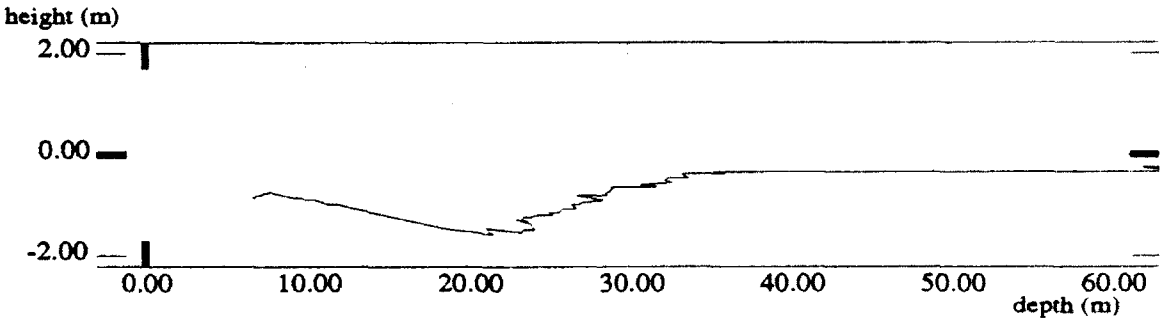


Fig. 55: Profile of depth at column 360 of "Field" data set

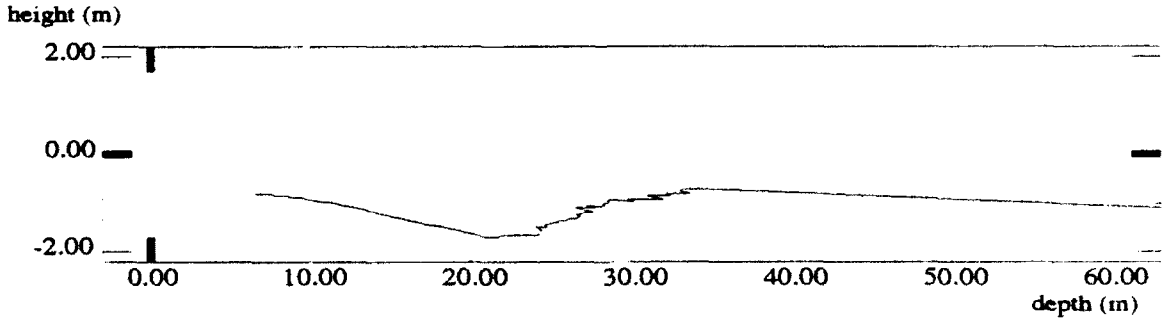


Fig. 56: Profile of depth at column 480 of "Field" data set

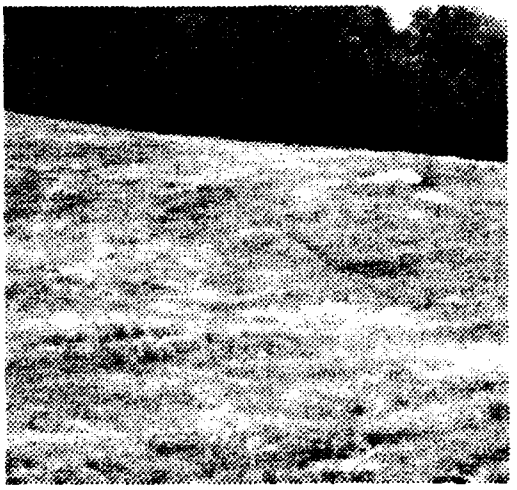


Fig. 57: "Hill" data

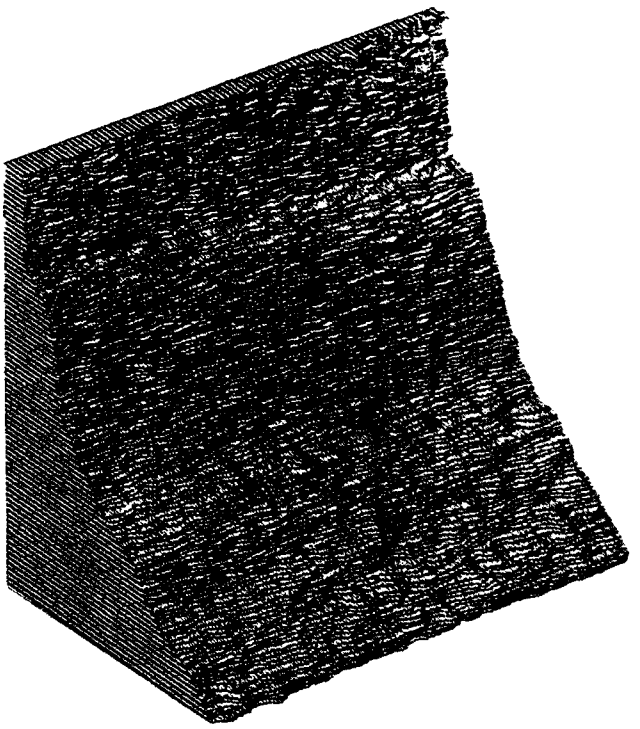


Fig. 58: Isometric plot of disparity ("Hill")

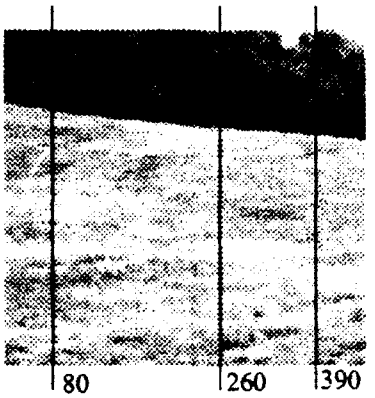


Fig. 59: Locations of profile of "Hill" data set

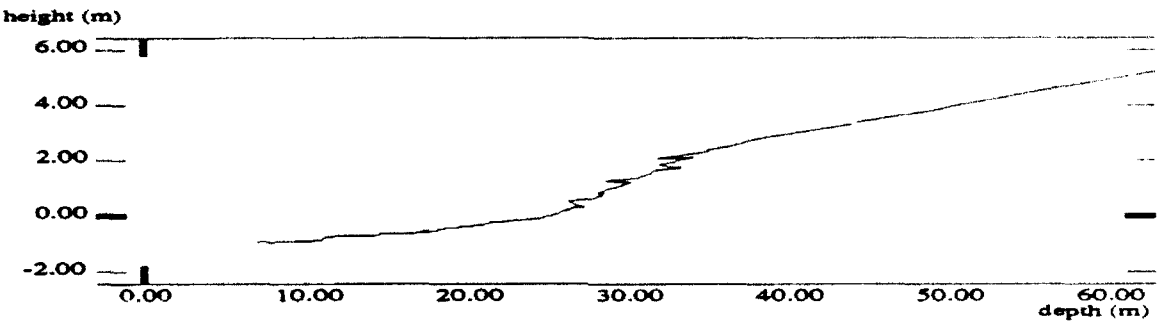


Fig. 60: Profile of depth at column 80 of "Hill" data set

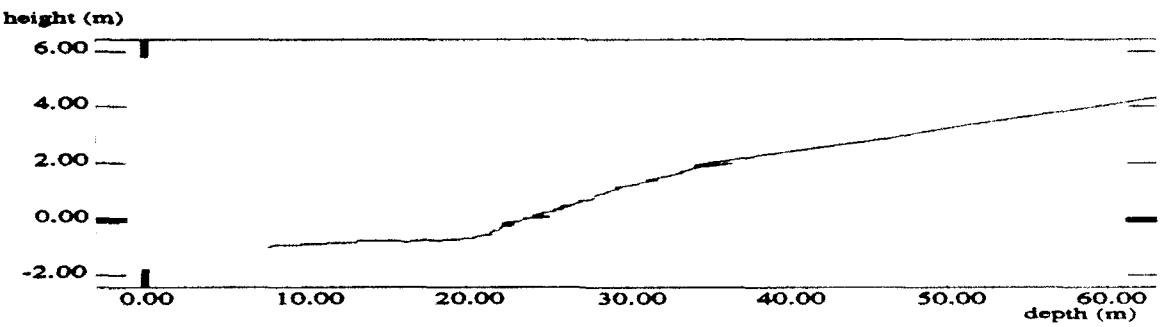


Fig. 61: Profile of depth at column 260 of "Hill" data set

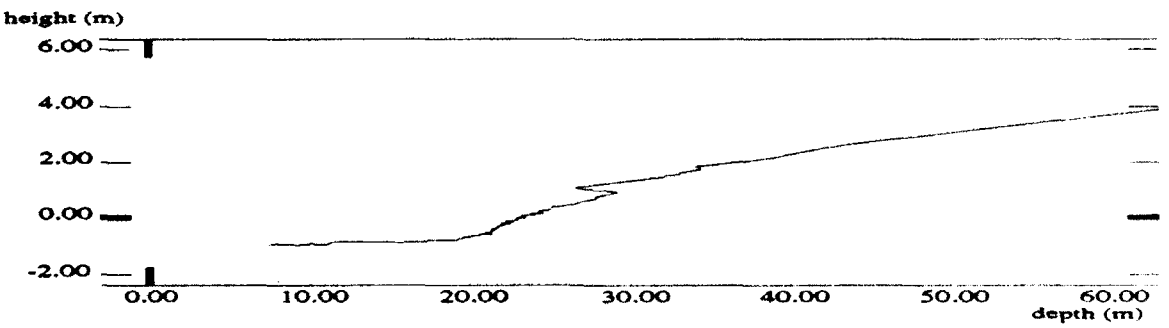


Fig. 62: Profile of depth at column 390 of "Hill" data set

Fig. 63: Detected false matches with type O

Fig. 64: Detected false matches with type S

Fig. 65: Detected false matches with type X

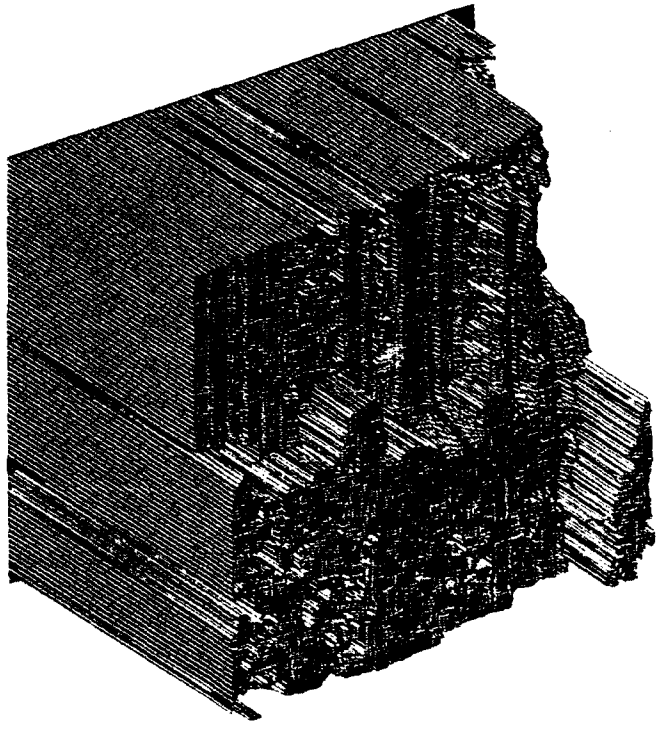


Fig. 66: Isometric plot of depth with good matches ("Parking meters")

Fig. 67: Detected false matches with type O

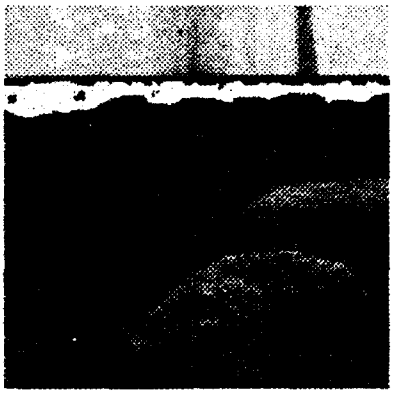


Fig. 68: Detected false matches with type S

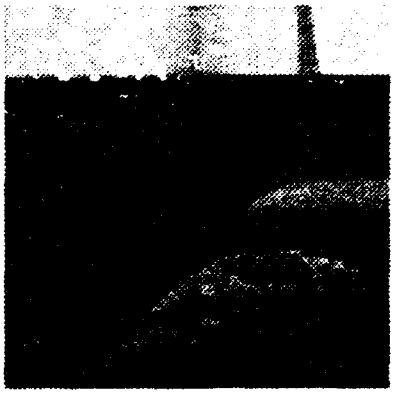


Fig. 69: Detected false matches with type X

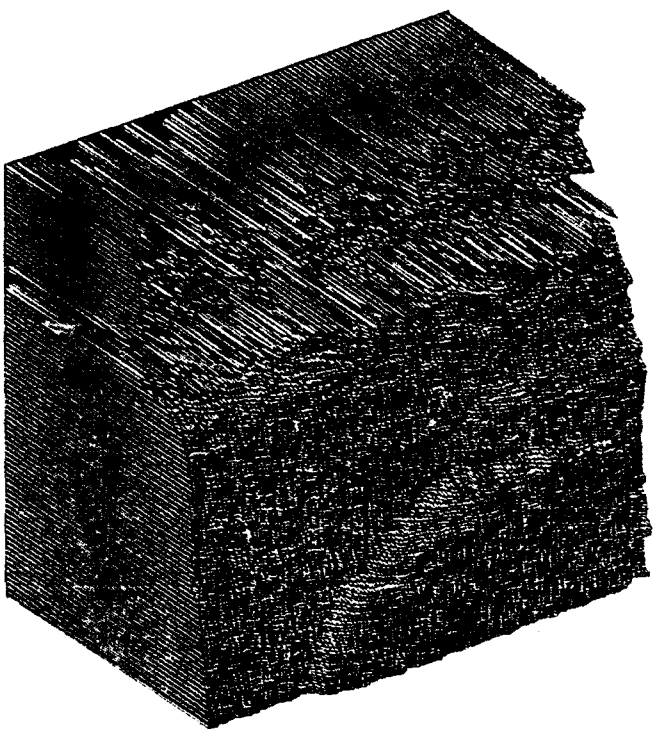


Fig. 70: Isometric plot of depth with good matches ("Sand")

Fig. 71: Detected false matches with type O

Fig. 72: Detected false matches with type S

Fig. 73: Detected false matches with type X

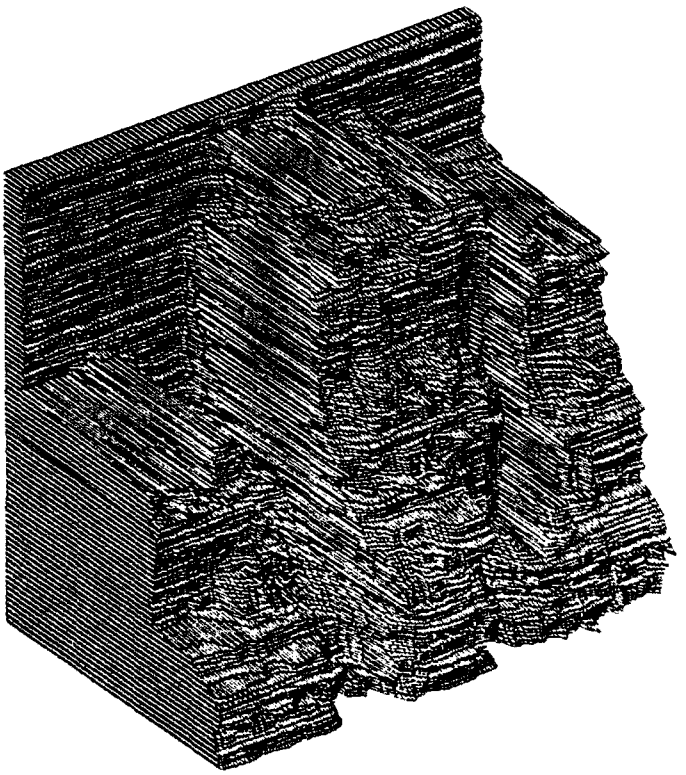


Fig. 74: Isometric plot of depth with good matches ("Train2")

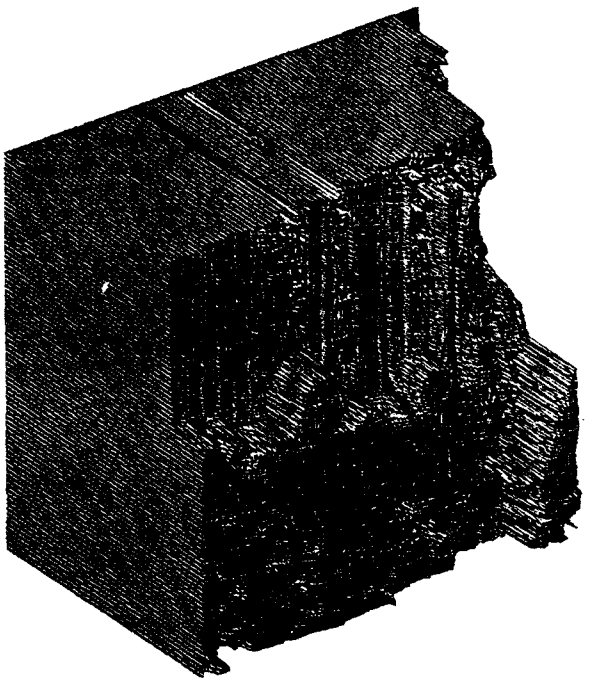


Fig. 75: Isometric plot of depth with corrected estimates ("Parking meters")

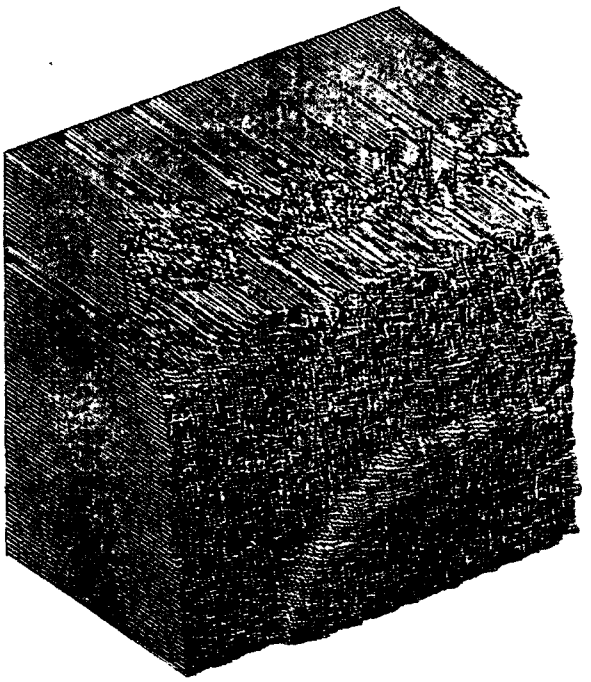


Fig. 76: Isometric plot of depth with corrected estimates ("Sand")

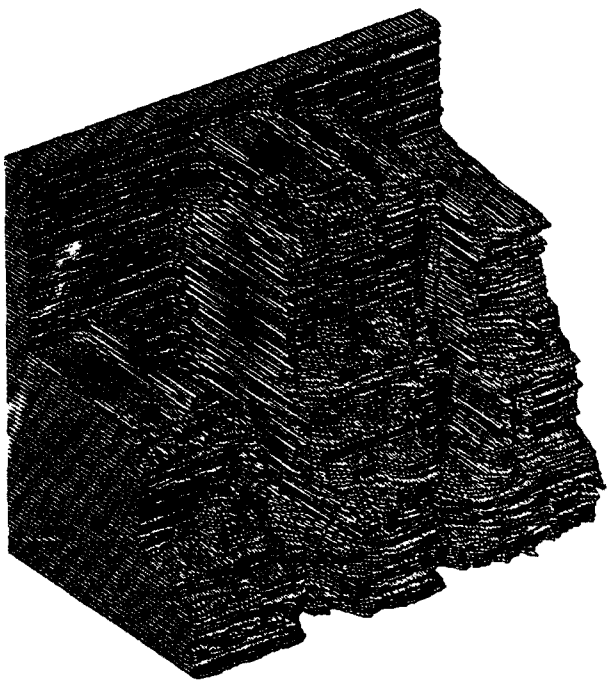


Fig. 77: Isometric plot of depth with corrected estimates ("Train2")